ROYAL INSTITUTE
OF TECHNOLOGY

Tree Detection and Species Identification using LiDAR Data

Mohammad Amin Alizadeh Khameneh

Master of Science Thesis in Geodesy No. 3127
TRITA-GIT EX 13-001

School of Architecture and the Built Environment
Royal Institute of Technology (KTH)
Stockholm, Sweden

January 2013

Abstract

The importance of single-tree-based information for forest management and related industries in countries like Sweden, which is covered in approximately 65% by forest, is the motivation for developing algorithms for tree detection and species identification in this study. Most of the previous studies in this field are carried out based on aerial and spectral images and less attention has been paid on detecting trees and identifying their species using laser points and clustering methods.

In the first part of this study, two main approaches of clustering (hierarchical and K-means) are compared qualitatively in detecting 3-D ALS points that pertain to individual tree clusters. Further tests are performed on test sites using the supervised k-means algorithm in which the initial clustering points are defined as seed points. These points, which represent the top point of each tree are detected from the cross section analysis of the test area. Comparing those three methods (hierarchical, ordinary K-means and supervised K-means), the supervised K-means approach shows the best result for clustering single tree points. An average accuracy of 90% is achieved in detecting trees. Comparing the result of the thesis algorithms with results from the DPM software, developed by the Visimind Company for analysing LiDAR data, shows more than 85% match in detecting trees.

Identification of trees is the second issue of this thesis work. For this analysis, 118 trees are extracted as reference trees with three species of spruce, pine and birch, which are the dominating species in Swedish forests. Totally six methods, including best fitted 3-D shapes (cone, sphere and cylinder) based on least squares method, point density, hull ratio and slope changes of tree outer surface are developed for identifying those species. The methods are applied on all extracted reference trees individually. For aggregating the results of all those methods, a fuzzy logic system is used because of its good reputation in combining fuzzy sets with no distinct boundaries. The best-obtained model from the fuzzy system provides 73%, 87% and 71% accuracies in identifying the birch, spruce and pine trees, respectively. The overall obtained accuracy in species categorization of trees is 77%, and this percentage is increased dealing with only coniferous and deciduous types classification. Classifying spruce and pine as coniferous versus birch as deciduous species, yielded to 84% accuracy.

Sammanfattning

I bland annat den svenska skogsindustrin, där landets yta ät täckt av skog till 65%, är det av stor vikt att få fram information om skogen som är baserad på varje individuellt träd. Därför fokuserat denna studie på algoritmer för trädigenkänning samt för att artbestämma träd. De flesta tidigare studier inom området baseras från flygfoton eller satellitbilder mindre fokus har lagts på metoder som andvänder punktmoln från laserscanning.

I första delen av studien görs en kvalitativ jämförelse av två olika sätt att arbeta med klusterbildning (hirarkisk och K-means) här söks efter 3-D ALS punkter som bildar individuella trädkluster. Fler tester utförs med ”supervised K-means”-algoritmen där de initierande klusterpunkterna definieras som seed-punkter. Dessa punkter som representerar förälder-noden i varje träd kommer från ”cross section” analys av testytan. När man jämför dessa metoder (hriarkisk, vanlig K-means och ”supervised K-means”) visar ”supervised K-means” bästa resultatet för att ta fram kluster för enstaka träd. Medelnoggrannheten är 90 % för att identifiera enstaka träd. Om man jämför resultatet från denna studie med DPM mjukvara, som utvecklats av Visimind för att göra analyser av LiDAR-data så har resultatet från den 85 % noggrannhet.

Den andra delen i studien består av att identifiera vilka arter träden har. För att kunna utföra analysen togs 118 olika träd ut för att användas som referensobjekt med arterna gran, tall och björk, de tre mest dominerande arterna i svenska skogar. Totalt användes sex olika metoder för att artbestämma träden, ”best fitted 3D shapes” (konisk, sfärisk och cylindrisk), minstakvadratmetoden, punkt densitet, ”hull ratio”, förändring i lutning för ytterytan. Dessa metoder användes sedan på alla referensträd individuellt. För att kunna aggregera ihop resultaten användes ”fuzzy logic”-system eftersom systemet har bra rykte vad det gäller att kombinera ”fuzzy sets”. Den bästa modellen om man ser till ”fuzzy”-systemet ger 73, 87 och 71 % noggrannhet då man identifierar respektive gran, tall och björk. Noggrannheten för alla sammanslaget för att kategorisera träd är 77 %, den procenten ökar då man väljer att klassificera barr och lövträd istället för mer artspecifikt, då får man istället 84 % noggrannhet.

Acknowledgment

Studying in KTH has been of great pleasure for me. My biggest thanks and highest respect go to my supervisor, Docent Milan Horemuž for his constant supervision and support during this thesis work. Without his encouragement, perhaps this thesis would never have been executed.

I am also thankful to Dr Krzysztof Gajdamowicz for his original idea for starting this study and the provided data by his company (Visimind AB), which made my thesis work more practical.

My sincere thank goes to Professor Lars Sjöberg for his comments on my thesis report.

I would like to appreciate all teachers, instructors, and staff in the Geodesy and Geoinformatic divisions for their high quality education and support.

I wish to thank all my classmates in KTH for their friendship and unforgettable time that we have had together during our studies; I am also indebted to Johanna Löfquist for her help in writing Swedish abstract.

Finally, I would like to appreciate my beloved parents for their unconditional love and support and special thank goes to my sister for her encouragement and moral support all the way through these years.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2-D	two-Dimensional
3-D	three-Dimensional
ALS	Airborne Laser Scanning
CIR	Colour Infra-Red
DCHM	Digital Crown Height Model
DSM	Digital Surface Model
DTM	Digital terrain Model
FIS	Fuzzy Inference System
FOV	Field of View
GIS	Geographic Information System
GPS	Global Positioning System
GUI	Graphical User Interface
LiDAR	Light Detection and Ranging
MF	Membership Function
nDSM	normalised Digital Surface Model
WPGMA	Weighted pair-group method using arithmetic average

1 Introduction

1.1 Background

Forests and trees are a crucial part of life on Earth, from maintaining biodiversity and cleaning the air and water, to providing basic human needs and contributing to culture and recreation[1]. Forests are important for us in three general aspects of environmental value, economic value and enjoyment value. They play an important role in the environment such as being a habitat for biodiversity, climate control and atmosphere purification. In addition to their role in global ecosystem, forests are the main source of timber and non-timber productions for the industry. The mentioned introduction about the importance of forests in human life is a motivation for doing investigations on forest inventory.

General characterization of the forest in terms of tree numbers, species, forest condition, and regeneration is called forest inventory. In other words, the forest inventory is a systematic collection of data and forest information for assessment or analysis. *The aim of the statistical forest inventory is to provide comprehensive information about the state and dynamics of forests for strategic and management planning*[2]. The inventory of trees has a history, which began in the late 18th century. The first inventories such as estimating the volume and dispersing of trees were carried out based on visual inspections. As the 20th century progressed, new statistical methods of sampling were established and the appearance of the new computer technologies as well as aerial images, opened a new era in this field.

These days, for collecting inventory data of an area, different approaches are being used with more advanced technologies such as Colour Infra-Red (CIR) images, aerial photographs and airborne laser data. The forest inventory is being performed at different resolutions, to gather the forest attributes for different purposes. In the diverse forests, a stand-wise approach is usually not sufficient for forest management planning as established in a number of European countries (Koch et al., 2006). The forest planning systems, especially for harvest management plans, typically work at the single tree level (e.g. Lämås & Eriksson, 2003). Therefore, single tree detection and related information extraction seems to become a prerequisite to fulfil these needs. Since remotely sensed data emerged and became a popular data source in forestry, there have been efforts to classify forest types of large areas (Nelson et al., 1984). The access of new age high resolution remote sensing data has facilitated users analysis in forest inventory field (Brandtberg & Warner, 2006).

Among high resolution remote sensing techniques, airborne laser scanning (ALS) has gained an important recognition as a complement for information extraction at individual tree level than other remote sensing sources (Koukoulas & Blackburn, 2005; Magnusson, 2006; Maltamo et al., 2006).

[1] http://www.janegoodall.ca/planet-releaf/documents/WhyForestsareImportant.pdf [Accessed 10 January 2013]
[2] http://en.wikipedia.org/wiki/Forest_inventory [Accessed 10 December 2012]

A short description about Light Detection and Ranging (LiDAR) technologies and products have been given by Wang (2009).

Since the last decade, the usage of three-dimensional (3D) ALS data with the application of different algorithms for single tree extraction is commonly exploited in the field of forestry in order to minimise the traditional forest inventory practices, which are very time, manpower and cost consuming. The costs of ALS data acquisition for single tree detection methods are higher compared to area based estimations (Næsset, 2002; Packalén, 2009). In addition, ALS based tree detection limited in a way that it can miss a portion of the smallest and/or understory trees (Persson et al., 2002). On the other hand, trees that are eventually detected correspond to the dominant tree layer (Vauhkonen, 2010).

As only the upper part of the tree crowns are visible in vertical aerial photos, the exact measurement of the tree crowns is not possible. On the other hand, the tree trunk is invisible and there is no possibility for direct stem diameter measurement. Based on the relation of crown size and the diameter of the stem, some regression models are created as allometric estimation of trees. Therefore using only aerial photos as a source of fundamental information for indirect estimation of tree species and tree allometry may result in prominent biases (Korpela, 2004). The demand for airborne LiDAR data with high quality (e.g. desirable footprint size, high point density) and more information (intensity, pulse width, number of echoes from each emitted laser pulse) has increased for various applications, like for the estimation of biophysical parameters in forest management performance using different techniques (Woodget et al., 2007; Suárez et al., 2008; Maltamo et al., 2009; Ørka et al., 2009) and environmental planning practices (Nilsson, 1996).

In addition to tree detection, the classification of trees is also involved in this thesis work. The classification of vegetation, especially trees, has been a piece of useful information for many studies, but it is a challenging task because remotely sensed imagery data, provides little information about the internal structures of tree canopies. In many studies, tree classifications are performed by human interpretation using aerial photos. The introduction of small footprint airborne LiDAR opened up many research possibilities for forest studies because of the capability of LiDAR to penetrate canopies vertically and revealed some of their internal structures, thus, providing geometrical information about tree crowns and boles. For that reason, it is logical to develop methodologies that include the internal structures of individual trees (Ko et al., 2009). In addition to internal structure of tree, the outer surface and shape of it can be evaluated by 3D geometrical shapes.

The major goal of this thesis work is to detect trees and identify their corresponding species. For this purpose, some algorithms are developed for detection and then based on clustering methods, the tree points are extracted and finally fuzzy logic inference system helps us to obtain a model for species identification. The thesis report is written in four major sections in which the Introduction section explains the overviews of methods that have been used in this study; the second section consists of the methodology of all developed algorithms. In the third part of the report, the results of the thesis algorithms are presented and compared with the results of other software for tree detection. The conclusion and future work for this study are outlined in the last section.

1.2 Overview of clustering methods

Clustering of objects is as ancient as the human need for describing the salient characteristics of men and objects and identifying them with a type. Therefore, it embraces various scientific disciplines: from mathematics and statistics to biology and genetics, each of which uses different terms to describe the topologies formed using this analysis. From biological "taxonomies", to medical "syndromes" and genetic "genotypes", the problem is identical: forming categories of entities and assigning individuals to the proper groups within it (Rokach & Maimoon, 2005). In the other words, cluster analysis divides data into groups (clusters) that are meaningful and/or useful. Classes or conceptually meaningful groups of objects that share common characteristics, play an important role in how people analyse and describe the world. Indeed, human beings are skilled at dividing objects into groups (clustering) and assigning particular objects to those groups (classification). Therefore, from a statistical pattern recognition view, clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) (Gupta, 2010).

Since clustering is the grouping of similar objects, some sort of measure that can determine whether two objects are similar or dissimilar is required. There are two main types of measures used to estimate this relation: distance measures and similarity measures. Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two objects x_i and x_j in the data set S as: $d(x_i, x_j)$. A valid distance measure should be symmetric and obtains its minimum value (usually zero) in case of identical objects. The distance measure is called a metric distance measure if it also satisfies the following properties:

1. Triangle inequality $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k) \quad \forall x_i, x_j, x_k \in S.$ (1-1)
2. $d(x_i, x_j) = 0 \rightarrow x_i = x_j \quad \forall x_i, x_j \in S.$ (1-2)

Given two *p*-dimensional objects that are characterized by a set of *p* measured attributes (variables), $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $x_j = (x_{j1}, x_{j2}, \dots, x_{jp})$. The distance between the two objects can be calculated using the Minkowski metric (Han & Kamber, 2001):

$$d(x_i, x_j) = \left(|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + \cdots + |x_{ip} - x_{jp}|^g \right)^{1/g} \quad (1\text{-}3)$$

The commonly used Euclidean distance between two objects is achieved when $g = 2$. Given $g = 1$, the sum of absolute paraxial distances (Manhattan metric) is obtained, and with $g = \infty$ one gets the greatest of the paraxial distances.

An alternative concept to that of the distance is the similarity function $s(\boldsymbol{x_i}, \boldsymbol{x_j})$ that compares the two vectors $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. This function should be symmetrical, i.e. $s(\boldsymbol{x_i}, \boldsymbol{x_j}) = s(\boldsymbol{x_j}, \boldsymbol{x_i})$, and have a large value when $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are somehow "similar" and constitute the largest value for identical vectors.

A list of clustering methods is shown in Figure 1, and common methods are explained further.

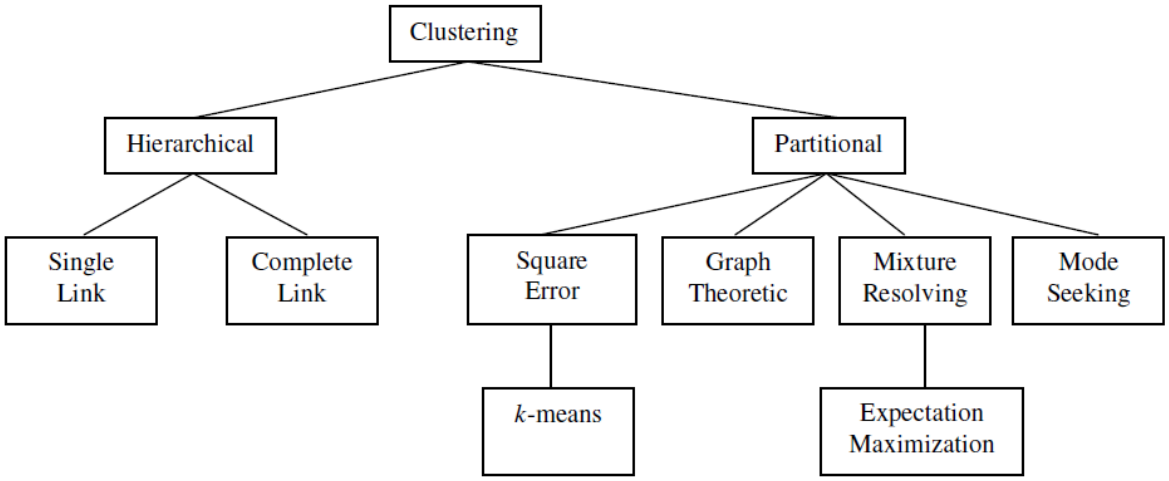


Figure 1: Different methods of clustering (adapted from Jain et al., 1999).

1.2.1 Hierarchical methods

The hierarchical methods construct the clusters by recursively partitioning the objects in either a top-down or bottom-up fashion. These methods can be subdivided as follows:

- *Agglomerative hierarchical clustering:* Each object initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is obtained.
- *Divisive hierarchical clustering:* All objects initially belong to one cluster. Then the cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained.

The result of the hierarchical methods is a dendrogram, representing the nested grouping of objects and similarity levels at which groupings change. A clustering of the data objects is obtained by cutting the dendrogram at the desired similarity level. The merging or division of clusters is performed according to some similarity measure, chosen so as to optimize some criterion (such as a sum of squares). The hierarchical clustering methods could be further divided according to the manner that the similarity measure is calculated (Jain et al., 1999):

- *Single-link clustering* (also called the connectedness, the minimum method or the nearest neighbour method)
- *Complete-link clustering* (also called the diameter, the maximum method or the furthest neighbour method)
- *Average-link clustering* (also called minimum variance method)

Generally, hierarchical methods are characterized with the following strengths:

- *Versatility:* The single-link methods, for example, maintain good performance on data sets containing non-isotropic clusters, including well separated, chain-like and concentric clusters.

- *Multiple partitions:* hierarchical methods produce not one partition, but multiple nested partitions, which allow different users to choose different partitions, according to the desired similarity level. The hierarchical partition is presented using the dendrogram.

The main disadvantages of the hierarchical methods are:

- The time complexity of hierarchical algorithms is at least $O(m^2)$ (where *m* is the total number of objects), which is non-linear with the number of objects.
- Hierarchical methods can never undo what was performed previously. Namely, there is no back-tracking capability.

1.2.1.1 Hierarchical clustering in MATLAB

The statistic toolbox in MATLAB provides a bunch of pre-defined functions for applying on data to do hierarchical clustering. In this thesis work, analysis of data set is carried out based on agglomerative hierarchical clustering. The algorithm can be described as below:

1. In first step, the similarity of pairs of objects in data set should be found. As the most popular way to evaluate similarity is the use of distance, and on the other hand, the most widely used distance measurement is Euclidean distance. The distances are calculated by *pdist* function, which uses Euclidean distance.
2. In next step, the pairs of points in close proximity are linked together by *linkage* function. This function uses the computed distances from previous step to determine the closeness of points to each other. The points are paired into binary clusters and those small clusters are grouped to bigger clusters until making a hierarchical tree that can be illustrated by *dendrogram* function.
3. In last step, the branches (leaves) of hierarchical tree should be pruned to get optimal number of clusters.

The principles of hierarchical clustering is explained below, which can be reached out using built in functions in MATLAB.

Similarity Measures

As explained already, the popular method for similarity measure is distance. Pairwise distance between pairs of objects can be determined by the *pdist* function with applying different methods, the famous and frequently used method is *Euclidean Distance* that can be defined as: $d = \sqrt{(\boldsymbol{x_s} - \boldsymbol{x_t}) * (\boldsymbol{x_s} - \boldsymbol{x_t})'}$ in which $\boldsymbol{x_s}$, $\boldsymbol{x_t}$ are vectors consisting of objects. For a data set made up of *m* objects, there are *m**(*m* – 1)/2 pairs in the data set. The other methods are: *Standardize Euclidean Distance (seucldiean), cityblock, minkowski, cosine, correlation …* (MATLAB documentation). In this thesis, the default option of the *pdist* function, *Euclidean Distance,* is used due to its simplicity and efficiency in dense data.

Linkages and Dendrograms

The linkage or amalgamation method is used to determine whether two clusters are sufficiently similar to be linked together. Amongst the different linkage methods, average linkage is widely used because *it is a compromise between the sensitivity of complete-link clustering to outliers and the tendency of single-link clustering to form long chains* (Manning et al., 2008). For the creation of a hierarchical cluster tree, the weighted average distance algorithm is used, or also known as **w**eighted **p**air-**g**roup **m**ethod using arithmetic **a**verages (WPGMA), one type of agglomerative or bottom-up algorithm (Sneath & Sokal, 1973) was used.

A dendrogram can be considered as a graphical interface of linkages. In Figure 2, the component of the dendrogram is shown. Dendrograms consists of many U-shaped lines connecting objects in the hierarchical tree and defines *links*. The horizontal axis in the dendrogram represents the indices of objects in the data set and the vertical axis shows the distances between the grouped objects (clusters). These distances can be interpreted as height of the links, which connects clusters to each other. Each node in the diagram represents one object if the total number of objects does not exceed 30, otherwise each node may represent more nodes (MATLAB documentation).

Verify the cluster tree

After linkage, the verification should be performed to see how accurate the computed distances and grouped objects are in comparison with the real distances and clusters. For this purpose, further investigations for verification analysis are carried out as follows:

- *Verify Dissimilarity:* in the hierarchical tree, every pair of objects is connected together in some level and the height of link shows the distance of clusters containing those two objects. This distance is also called *cophenetic distance* between two objects. To reach to our purpose, the computed distances by *pdist* are compared with cophenetic distances, which are a result of linkage. If the clustering is valid, the linking of objects should have a strong correlation with distances. The value of the cophenetic correlation coefficient brings the answer. The closer the value to 1, the more accurate the reflected data. As explained before, the *pdist* and *linkage* functions have different methods to apply on data, so those differences affect the cophenetic coefficient, it is clear that the method, which gives better cophenetic value, will be used in process (MATLAB documentation). Equation (1-4) shows the mathematical definition of cophenetic coefficient:

$$c = \frac{\sum_{i<j}(Y_{ij}-y)(Z_{ij}-z)}{\sqrt{\sum_{i<j}(Y_{ij}-y)^2 \sum_{i<j}(Z_{ij}-z)^2}} \quad (1\text{-}4)$$

  where:
  - C is cophenetic correlation coefficient.
  - Y_{ij} is the Euclidean distance between objects i and j.
  - Z_{ij} is the cophenetic distance between objects i and j.
  - y and z are the averages of Y and Z, respectively.

- *Verify Consistency:* one of the ways to determine the cluster divisions in a data set is comparing the height of the link with the height of neighbour links underneath that link. If the link has approximately the same height of below links, then it is called to have *consistency* with components and there is no division between clusters. The value of consistency would be zero if two nodes have been investigated under one link. On the other hand, if the link height has noticeable difference with the height of the below links, it shows a natural division among the data set in that level of hierarchical tree. That link is said to be *inconsistent* with the links below it. The relative consistency of each link in a hierarchical cluster tree can be expressed as the *inconsistency coefficient*. This value compares the height of a link with the average height of the links below it. Links that join distinct clusters have a high inconsistency coefficient; links that join indistinct clusters have a low inconsistency coefficient (MATLAB documentation).

Create Clusters

For separating clusters using the dendrogram, there are several choices. As explained before, one of the factors for cutting the tree is the inconsistency coefficient, which presents a relative value of consistency between links in different levels of a tree. The second factor is the height of links. As shown in Figure 2, any horizontal line can cut the dendrogram in specific height for obtaining the desired number of clusters. For instance, the drawn dashed line cuts the tree to six clusters. The function of *cluster* can manage this task.

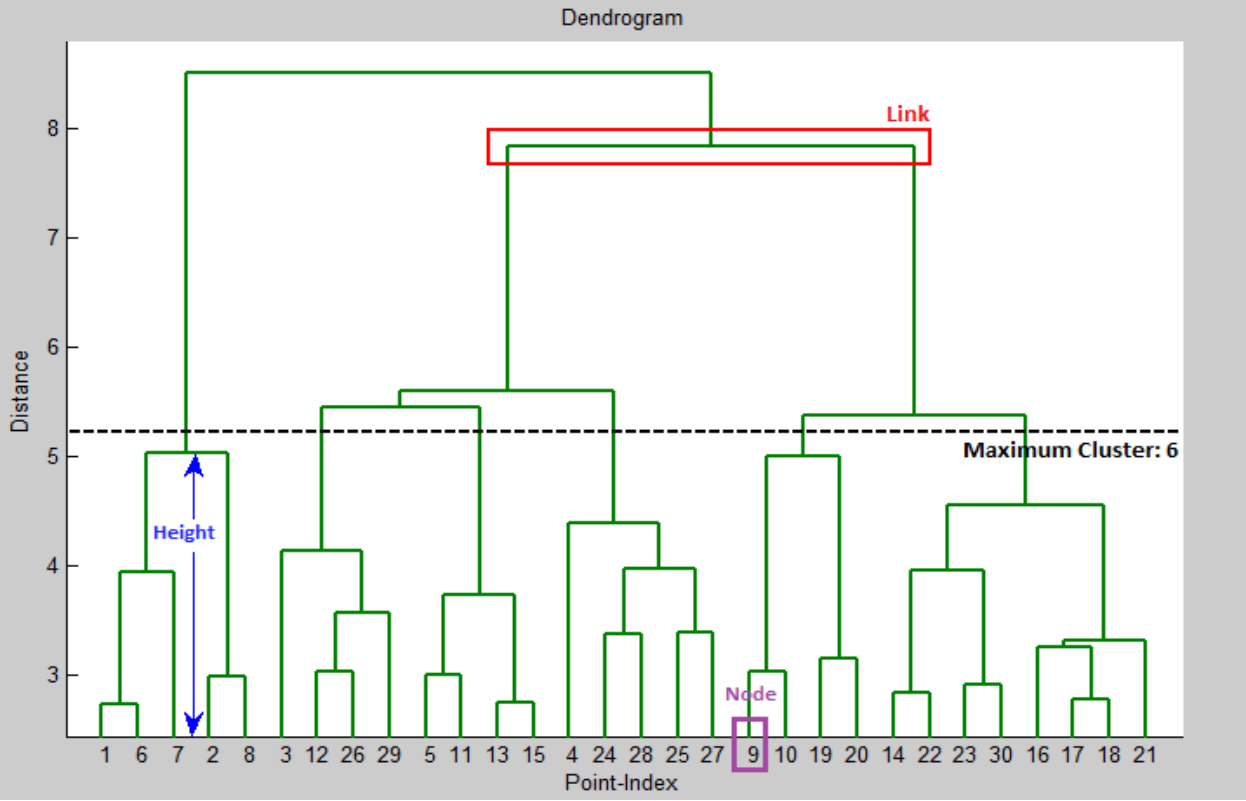


Figure 2: The Dendrogram and its components.

1.2.2 Partitioning methods

The workflow for the partitioning methods is relocating objects by moving them from one cluster to another, starting from an initial partitioning. For running these methods, the number of clusters should be determined and set in advance. A relocation method iteratively relocates points between the *K* numbers of clusters. Error minimisation algorithms are one of the sub-sections of partitioning methods, which are the most intuitive and frequently used methods among other types. The basic idea is to find a clustering structure that measures the *distances* of each object to its representative value. The most well-known criterion is minimising the *Sum of Squared Errors* (SSE), which minimises the total squared Euclidean distance of objects to their representative values (Rokach & Maimoon, 2005). The most common algorithm using squared error is *K-means*.

1.2.2.1 K-means

The *K-means* clustering is an iterative partitioning-based clustering mechanism. *K-means* has become the most common technique for partitioning a dataset in which the sum of the within-cluster variances are minimised (McQueen, 1967). The algorithm partitions the data set to *K* clusters ($C_1, C_2, C_3, \dots, C_K$), represented by centroids. The centre point of each cluster is the mean of all objects, which it belongs to.

The algorithm starts with an initial set of cluster centers, chosen at random or according to some heuristic procedure. In every iteration, each object is assigned to its nearest cluster centre according to the Euclidean distance between them. In the next step, the cluster centers are re-computed. The iterations continue while either partitioning error is not reducing by centre relocation, which means that the obtained cluster is the optimal one, or the iteration numbers reach to pre-defined number of iterations (Rokach & Maimoon, 2005).

The popularity of this method is because of its linear complexity, which is considered as an advantage versus other clustering methods like hierarchical clustering with non-linear complexity. The complexity of *K-means* for *l* number of iterations and *K* cluster for m number of objects is $O(Klm)$. In addition, the ease of interpretation, simplicity of implementation, speed of convergence and adaptability to sparse data are other advantages for this algorithm (Dhillon & Modha, 2001).

1.2.2.2 K-means in MATLAB

The statistics toolbox in MATLAB has functions to perform two types of clustering, hierarchical clustering, which was explained before and k-means clustering. It has been mentioned already that K-means treats observations in the data set as objects, which have location and distance; it partitions the objects to K clusters such that all objects within one cluster are as close to each other as possible and far from other objects in other clusters. This approach can be applied on data by the *kmeans* function in MATLAB, this function has a capability to start the clustering either with random initial points or pre-set initial points and the method for computing the distances can be defined as well as the number of iterations (MATLAB documentation).

Validation of clustering

After clustering the data by using any technique such as K-means into K clusters, *silhouette* plot can display a measure of how close each point in one cluster is to points in the neighbouring clusters. Peter Rousseeuw first described silhouette plot, which is a measure of cluster goodness in statistical analysis. A silhouette value ($s(i)$) for each object (i) is defined as follows (Rousseeuw, 1987):

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{1-5}$$

where

- $a(i)$ is the average dissimilarity of object (i) with all other objects inside cluster A that object (i) belongs to.
- $b(i)$ is the average dissimilarity of object (i) with all other objects from nearest neighbouring cluster to A.

From equation (1-5), the range of $s(i)$ is deducted as: $-1 \leq s(i) \leq 1$. If the numeric value of silhouette closes to 1, it implies that "*within*" dissimilarity $a(i)$ is much smaller than the smallest "*between*" dissimilarity $b(i)$, therefore we can say that object (i) is *well-clustered*. If silhouette value closes to zero, it shows that the value of $a(i)$ and $b(i)$ are somehow the same and there is no distinct conclusion that which cluster the object (i) belongs to, so it can be considered as *intermediate case*. The worst case is to get silhouette value close to -1, which means that object (i) belongs to other cluster and "*misclassified*" (Rousseeuw, 1987).

The silhouette value, which is used for analysing the clustering, is the mean of all the silhouettes of clusters. Figure 3 illustrates one sample silhouette plot for a data set with three clusters. It is clear that the third cluster has less dissimilarity within the cluster and totally better partitioned.

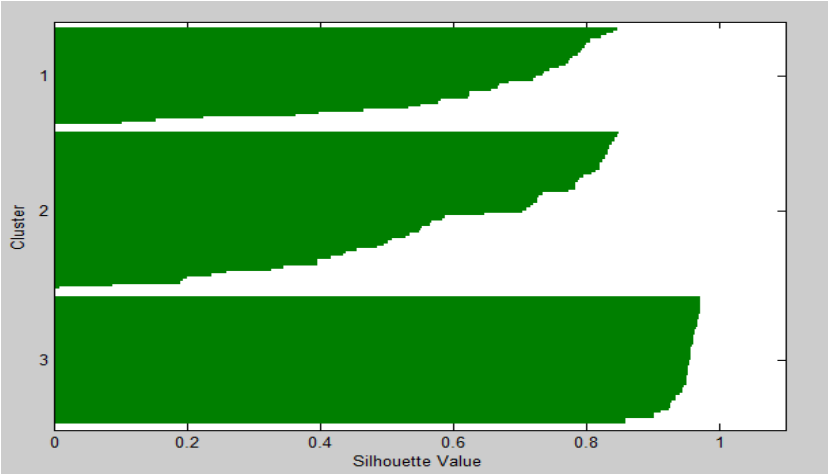


Figure 3: Sample silhouette plot shows the values in horizontal axis and the number of clusters in vertical axis.

1.3 Single tree detection

In the field of tree detection, some studies have been carried out since previous years for the purpose of either detecting a general number of trees in forest areas or as a pre-requisite procedure for investigating on tree species. In recent years, more or less all studies follow a specific method for tree detection in which the researchers mostly deal with data obtained from LiDAR, CIR (Colour InfraRed) or spectral images and orthophotos. In the study carried out by Koch et al. (2006), using Digital Terrain Model (DTM) and Digital Surface Model (DSM) prepared from LiDAR points, a Digital Crown Height Model (DCHM) was computed from subtraction of so-called DTM and DSM for each pixel, which represents the height value of canopy in each pixel. DCHM maps the surface of the canopy and for increasing the precision of detection, a Gaussian smoothing model is applied to DCHM. In the smoothed DCHM, treetops were determined based on the local maximum filter in which the pixel is counted as local maxima if the other nearest neighbour pixels have less height value than that pixel. This process started from the local maxima and extended as long as the low height value is existed in neighbour pixels. This approach, which is called Pouring algorithm, has a similar function as the classical watershed algorithm. Many researches on tree detection follow this approach, for instance, similar studies were carried out by Heinzel et al. (2008) and Reitberger et al. (2009).

The drawback of the so called Pouring approach goes back to interpolation, since the interpolation process smoothes the surface drastically, neighbouring trees cannot be separated properly and it is probable to become segmented as group of trees instead. The other problem is related to detecting short trees from DCHM. As small trees are dominated by larger trees in dense forests, it is impossible to find them by pixel analysis (Reitberger et al., 2009).
Tiede et al. (2005) performed tree detection using an algorithm based on a regression model, which was linking the crown-width to the tree-height for finding local maxima and used similar neighbouring methods for crown delineation.

Steps in tree detection procedure

In the first part of this thesis work, a tree detection process is performed using two different clustering methods. Their principle was explained in the previous section. The K-means algorithm (an iterative partitioning top-down approach) and agglomerative bottom-up hierarchical algorithm are two clustering methods used in this study, which help us to obtain any tree in the study area as one individual cluster. The K-means method is applied to data set in two forms: supervised and unsupervised. Then the clustering results of all these three types are compared with each other. Finally, for the verification of tree detection, the obtained result is compared with the result of *DPM* software, which is developed by *Visimind AB* Company for processing LiDAR data. The major steps for tree detection are:

- Determination of tree numbers in the study area
- Applying simple (unsupervised) K-means method
- Performing supervised K-means approach
- Applying hierarchical method

1.4 Overview of tree species classification

Sweden is a country dominated by forests, so any type of studies, which leads to attaining more reliable and useful information such as trees count, crown size, growth rate and species in this area would be helpful. Such information would be used by research institutions, forest owners, industry, governmental or nongovernmental organizations and sections, which are active in this field. As stated by Hyyppä et al. (2008) the information about tree species is of particular interest for forest applications. As a quick glance at the statistics of land use and tree species in Sweden, the content of Figure 4 would be useful. It is clear from the figure that the majority of land in Sweden belongs to forestial areas, which based on Swedish National Forestry Inventory website, is around 28 million hectares from total 41 million hectares of land area. The forest land in Sweden is covered mostly by three species of Norway spruce (*Picea abies*), Scots pine (*Pinus sylvestris*) and a small portion of birch (Betula spp.) trees. It should be mentioned that other type of trees are also existing like oak, aspen and other deciduous types but the overall percentage of them is not as much as other so called species. Automatic classification into these tree species groups would be useful for forest management planning and monitoring the environment (Holmgren et al., 2008). The second part of this thesis work focuses on distinguishing the tree species based on laser points. As explained already the dominant tree species are pine, spruce and birch so, the algorithms are developed to classify trees into these three species.

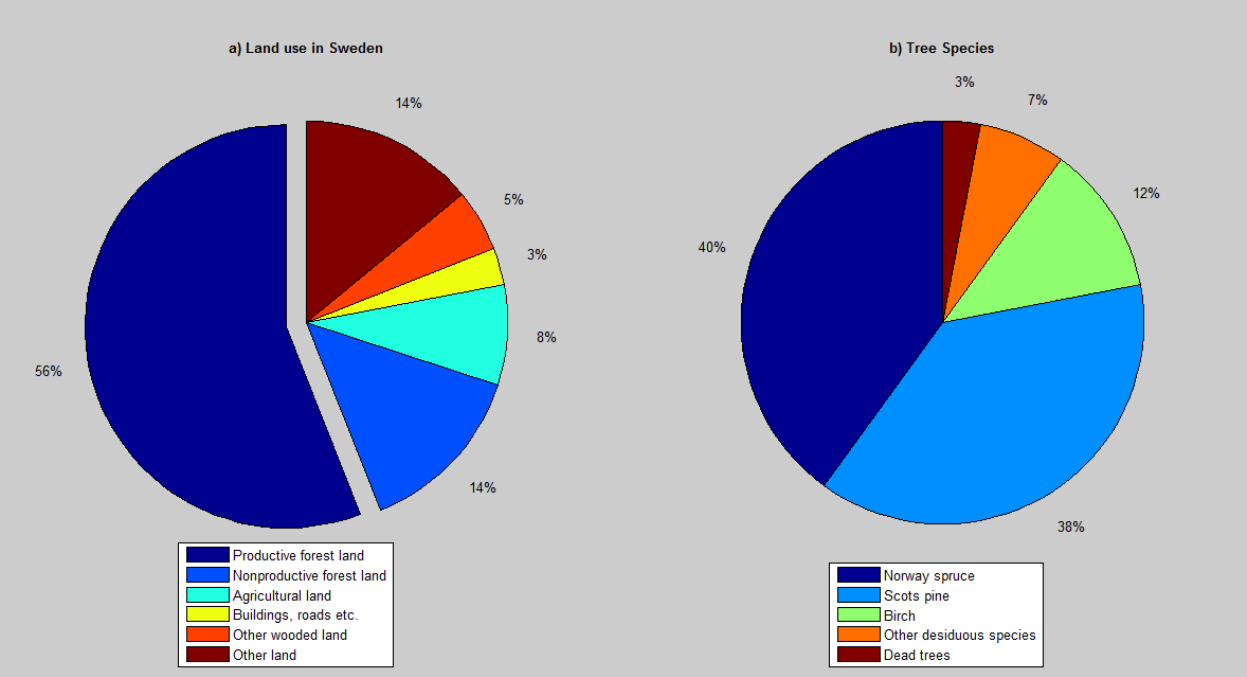


Figure 4: Chart (a) shows the statistics of land use in Sweden (total land area is 40.8 million ha) and chart (b) represents the percentage of tree species in productive forestland, data source is Swedish National Forestry Inventory (2004-2008). The land, which is producing or capable of producing commercial forest products, is called productive forestland.

Previous works on tree species identification

In the field of tree species detection, some projects have been carried out in the recent decade and some approaches as well as software are introduced in different articles and scientific journals. Different types of data were used for achieving the goal. Based on the geographic area that the data come from, the approaches and purpose for distinguishing tree species are changing. The studies in Northern European countries mostly focus on determining different coniferous trees, while the studies in central or southern parts are aimed to distinguish deciduous tree types. Due to the purpose of this study, most of the reviewed articles were investigating tree species problems in northern Europe. Some of the previous works are mentioned concisely below:

- Liang et al. (2007) and Reitberger et al. (2006) were distinguishing coniferous and deciduous trees with LiDAR based attempts. First study was performed under leaf-off conditions and they assumed that in coniferous trees, first and last pulse signals are reflected by tree tops while in deciduous trees, first pulse would hit the tree tops and last signals would reflect ground, based on this difference they obtained 89% accuracy in classifying coniferous-deciduous types. In second study, they used leaf-on data and obtained 80% accuracy for same classification.
- Ørka et al. (2007) used two intensity metrics of mean intensity and standard deviation of intensity. Those values were computed for echo categories in each tree. They achieved 68% to 74% accuracy in classifying tree species depending on the number of considered variables.
- Using combination of laser data and multi spectral images by Persson et al. (2006), yields quite good accuracy for classifying coniferous and deciduous trees.
- Heinzel et al. (2008) did an investigation on a test area in Poland using laser scanning data and CIR (Colour Infra-Red) aerial photographs for classification of oak, beech and coniferous tree types. They separated the spectral bands into near infrared, red and green and further transformed into hue, saturation and intensity channels; previously detected tree polygons are fitted to spectral data and based on comparing different channels, the species are being classified. The overall accuracy of 83% was achieved.
- In another study, Hollaus et al. (2009) achieved 83% accuracy in determining the species of spruce, larch and beech trees. Their approach uses geometric information such as echo width and backscatter cross section, extracted from full-wave form ALS data, for identifying tree species.
- Fusion of LiDAR data and multi-spectral aerial image has been studied for tree detection, measuring tree heights and for estimation of stand volume (e.g. Popescu et al. 2004). However, there are some works also carried out using this technique for species identification (e.g. Hill & Thomson, 2005; Koukoulas & Blackburn, 2005); the validation procedure of their study was carried out on an aggregated level, i.e. not on an individual tree level. Holmgren et al. (2008) performed tree species classification on an individual tree level. They used LiDAR data for segmenting tree crowns and combination of LiDAR data with DMC (Digital Mapping Camera) pixel values for segmented crowns for carrying out tree identification. Spectral values were extracted by projection of the LiDAR derived segments onto digital aerial images. They did quite a broad study on 1711 trees in this field, the achieved accuracy using just LiDAR data was 88%

while combining those data with high-resolution spectral images in autumn and summer, yields 96% accuracy for classifying three species of spruce, pine and deciduous trees in a test area situated in southern Sweden.

- The last reviewed literature refers to a study by Ko et al. (2009). They performed deciduous-coniferous classification for 65 trees using leaf-on single flight LiDAR data. Single trees were separated manually and the geometrical shape of the crown from real LiDAR data was compared with artificially generated point clouds by the Lindenmayer system language. Convex hull calculation and buffer analysis were also carried out for catching differences between those two types. The classification accuracy of their study is 85%-88%.

1.5 Fitting shapes based on least squares method

In the second part of this study, it is needed to get an estimation of how quadratic shapes can fit to an extracted point cloud of single trees. For this purpose, we found the least squares method efficient. As a general definition "*the least squares method is a standard approach to the approximate solution of overdetermined systems, i.e., sets of equations in which there are more equations than unknowns. Least square means that the overall solution minimises the sum of the squares of the errors made in the results of every single equation."* The most used application of this method is in data fitting. The best fit minimises the sum of the squared residuals, which is the difference between an observed value and the fitted value provided by a model. The least squares method is divided to a linear (ordinary) and a non-linear method; the linear approach, which is called linear regression in statistics, is minimising the sum of squared vertical distances of objects in a dataset from predicted linear approximation. Usually the linear least squares method is a closed-form solution while the non-linear approach follows an iterative refinement; at each iteration the system is approximated by a linear equation so the base of both methods are similar. It should be mentioned that for starting a non-linear algorithm, an initial estimation is needed (Wikipedia documents). The general equation of the least squares method can be expressed:

$$\sum_{i=1}^{n} \boldsymbol{\varepsilon}_i^2 = minimum \tag{1-6}$$

where $\boldsymbol{\varepsilon_i}$ denotes the estimated error or residual of observation l_i for $i = 1, 2, \dots, n$. The above condition applies when the observations are uncorrelated, otherwise the corresponding condition of the least squares method becomes:

$$\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = minimum \quad or \quad \boldsymbol{\varepsilon}^T C^{-1} \boldsymbol{\varepsilon} = minimum \tag{1-7}$$

where $\boldsymbol{\varepsilon}$ is the residual vector containing all residuals ε_i , which are derived together with the optimal estimate, $\boldsymbol{x}$ and C is the variance-covariance matrix of l (Fan, 2010).

As we deal with the 3D surface and model of trees, different geometrical shapes are used for fitting to tree shapes. These geometrical shapes (cylinder, cone and sphere) have non-linear equations. We need to linearize the non-linear observation equations of quadratic shapes to be able to use the least squares method for acquiring best fit shapes. The general linearization procedure is as follows (Fan, 2010):

It is assumed that we have n observation $l_1, l_2, \dots, l_n$ and the true value and error of l_i are $\tilde{l}_i$ and ε_i such that $\tilde{l}_i = l_i - \varepsilon_i$ $(1 \le i \le n)$. In addition, it is assumed that each $\tilde{l}_i$ is a non-linear function of m unknown parameters x_j $(1 \le j \le m)$:

$$\tilde{l}_i = f_i(x_1, x_2, \dots, x_m) \tag{1-8}$$

Let x_j^0 and δx_j denote an approximate value of x_j and its correction, such that $x_j = x_j^0 + \delta x_j$ $(1 \le j \le m)$:

$$\boldsymbol{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \boldsymbol{X^0} + \boldsymbol{\delta X}\,,\ \boldsymbol{X^0} = \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_m^0 \end{bmatrix},\ \boldsymbol{\delta X} = \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_m \end{bmatrix} \tag{1-9}$$

Expanding the equation (1-8) into a Taylor series, the linear observation equation is obtained:

$$\tilde{l}_i = l_i - \varepsilon_i = a_{i1}\delta x_1 + a_{i2}\delta x_2 + \cdots + a_{im}\delta x_m + c_i \tag{1-10}$$

where

$$a_{ij} = \frac{\partial f_i}{\partial x_j}\,;\quad c_i = f_i(x_1^0, x_2^0, \dots, x_m^0)\,;\quad i = 1, 2, \dots, n\,;\quad j = 1, 2, \dots, m \tag{1-11}$$

Hence, the linearized equation can be written in the general form of:

$$\boldsymbol{L} - \boldsymbol{\varepsilon} = \boldsymbol{A}\,\boldsymbol{\delta X} \tag{1-12}$$

where:

$$\boldsymbol{L} = \begin{bmatrix} l_1 - c_1 \\ l_2 - c_2 \\ \vdots \\ l_n - c_n \end{bmatrix},\ \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix},\ \boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix},\ \boldsymbol{\delta X} = \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_m \end{bmatrix} \tag{1-13}$$

The least squares solution of $\boldsymbol{\delta X}$ can be written as:

$$\boldsymbol{\delta \hat{X}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{L} \tag{1-14}$$

Finally, the least squares estimated unknown parameters as follows:

$$\boldsymbol{\hat{X}} = \boldsymbol{X}^0 + \boldsymbol{\delta \hat{X}} \tag{1-15}$$

The standard deviation of residuals, which is used as an assessment value for the fitted shapes in this study, is calculated as:

$$\hat{\sigma}_0 = \sqrt{\frac{\hat{\boldsymbol{\varepsilon}}^T\hat{\boldsymbol{\varepsilon}}}{n-m}} \qquad \text{where} \qquad \hat{\boldsymbol{\varepsilon}} = \boldsymbol{L} - \boldsymbol{A}\,\boldsymbol{\delta \hat{X}} \tag{1-16}$$

The standard errors of the unknowns are obtained from variance matrix of unknowns by:

$$\boldsymbol{C}_{\hat{X}\hat{X}} = \boldsymbol{C}_{\delta\hat{X}\delta\hat{X}} = \hat{\sigma}_0{}^2(\boldsymbol{A}^T\boldsymbol{A})^{-1} \tag{1-17}$$

For getting a better insight of how extracted reference tree points look like, three quadratic shapes (sphere, cone and cylinder) are fitted to the data set based on the least squares approach. The procedure of the least squares fitting of the shapes is described in the following subsections.

1.5.1 Fitting a sphere on a data set

LiDAR is a technology in which the scanning device sends a multitude of signals in a very short time, resulting in a point clouds containing thousands of data points. This technology has many applications such as monitoring construction sites, developing an as-built model of structures, etc. Point clouds could represent many 3D shapes of features but as mentioned before we deal with three shapes in this study, so the principle of fitting a sphere is reviewed in this section. The general equation of a sphere can be written as:

$$r^2 = (x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 \tag{1-18}$$

where

- r is the radius of the sphere.
- x_c , y_c and z_c are the coordinates of the centre point.
- x , y and z are the coordinates of a point on the surface of the sphere

Our goal is to use scanned points (x_i , y_i , z_i) to find the best-fit radius $\hat{r}$ and coordinates of the centre ($\hat{x}_c$, $\hat{y}_c$, $\hat{z}_c$), which fulfils the least squares condition:

$$\sum(r_i - \hat{r})^2 = \sum \varepsilon_i^2 = min \tag{1-19}$$

where

- $r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2}$; $i = 1, 2, \dots, n$ (1-20)
  - n is the number of points in the point cloud.

To be able to apply the linear model (1-12), we have to linearize Equation (1-20) by Taylor expansion around approximate coordinates of centre (x_{c0} , y_{c0} , z_{c0}):

$$r_i = r_{i0} + \frac{\partial r_i}{\partial x_c}|_{x_{c0}}\,\delta x_c + \frac{\partial r_i}{\partial y_c}|_{y_{c0}}\delta y_c + \frac{\partial r_i}{\partial z_c}|_{z_{c0}}\delta z_c \tag{1-21}$$

where

- $r_{i0} = \sqrt{(x_i - x_{c0})^2 + (y_i - y_{c0})^2 + (z_i - z_{c0})^2}$ (1-22)
- δx_c , δy_c and δz_c are the improvements to approximate coordinates so that:
  $\hat{x}_c = x_{c0} + \delta x_c$; $\hat{y}_c = y_{c0} + \delta y_c$; $\hat{z}_c = z_{c0} + \delta z_c$

With $r_i = r + \varepsilon_i$, we can rewrite Equation (1-21) in form of Equation (1-10) as:

$$r_{i0} - \varepsilon_i = r - a_i\delta x_c - b_i\delta y_c - c_i\delta z_c \tag{1-23}$$

where

- $a_i = \frac{\partial r_i}{\partial x_c}|_{x_{c0}} = \frac{-(x_i - x_{c0})}{\sqrt{(x_i - x_{c0})^2 + (y_i - y_{c0})^2 + (z_i - z_{c0})^2}}$
- $b_i = \frac{\partial r_i}{\partial y_c}|_{y_{c0}} = \frac{-(y_i - y_{c0})}{\sqrt{(x_i - x_{c0})^2 + (y_i - y_{c0})^2 + (z_i - z_{c0})^2}}$
- $c_i = \frac{\partial r_i}{\partial z_c}|_{z_{c0}} = \frac{-(z_i - z_{c0})}{\sqrt{(x_i - x_{c0})^2 + (y_i - y_{c0})^2 + (z_i - z_{c0})^2}}$

Equation (1-12) becomes:

$$\boldsymbol{L} = \begin{bmatrix} r_{10} \\ \vdots \\ r_{n0} \end{bmatrix} \; ; \; \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix} \; ; \; \boldsymbol{A} = \begin{bmatrix} 1 & -a_1 & -b_1 & -c_1 \\ & & \vdots & \\ 1 & -a_n & -b_n & -c_n \end{bmatrix} \; ; \; \boldsymbol{\delta X} = \begin{bmatrix} r \\ \delta x_c \\ \delta y_c \\ \delta z_c \end{bmatrix} \tag{1-24}$$

where x_{c0} , y_{c0} and z_{c0} denote the approximated mean coordinates of point cloud as an initial point for starting the iteration procedure of the least squares. They are being updated at the end of every iteration loop by $\delta\hat{x}_c$, $\delta\hat{y}_c$ and $\delta\hat{z}_c$.

The residual matrix ($\boldsymbol{\varepsilon}$) is computed as $\hat{\boldsymbol{\varepsilon}} = \boldsymbol{L} - \boldsymbol{A}\boldsymbol{\delta}\widehat{\boldsymbol{X}}$, which is used for computing standard deviation value for analysing the quality of fitting shape to point cloud by Equation(1-16). The standard errors of unknowns are also computed from Equation (1-17) as:

$$\sigma_{\hat{r}} = \sqrt{\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}}(1,1)}\,, \; \sigma_{\delta\hat{x}_c} = \sqrt{\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}}(2,2)}\,, \sigma_{\delta\hat{y}_c} = \sqrt{\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}}(3,3)}\,, \; \sigma_{\delta\hat{z}_c} = \sqrt{\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}}(4,4)} \tag{1-25}$$

where $\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}} = \hat{\sigma}_0{}^2(\boldsymbol{A}^T\boldsymbol{A})^{-1}$ is a variance matrix of unknowns by dimension of 4*4, and $\hat{\sigma}_0 = \sqrt{\frac{\hat{\boldsymbol{\varepsilon}}^T\hat{\boldsymbol{\varepsilon}}}{n-4}}$

Figure 5 shows the least squares fitted Sphere to a sample point cloud.

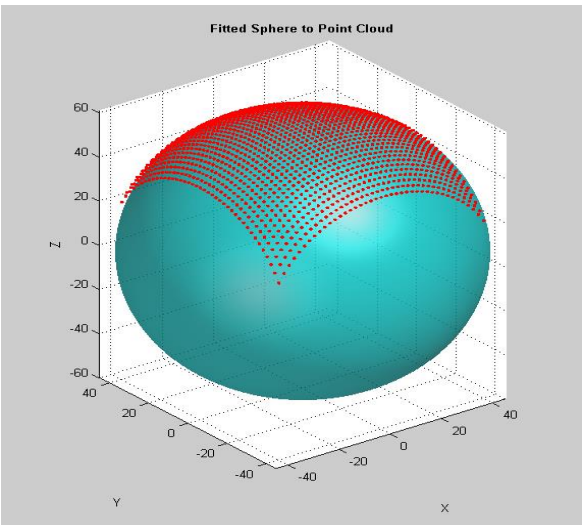


Figure 5: Fitted sphere to the point cloud. It should be mentioned that the point cloud (red points) is generated as a partial cloud of a sphere with radius of 40 units, origin at (0, 0, 0). As shown in the figure the algorithm successes to fit a sphere with those parameters.

1.5.2 Fitting a cone on a data set

Second quadric shape that is needed to fit to the point cloud is a cone, the procedure for performing the least squares method is the same as what has been performed for the sphere fitting, but the difference is in the general equation of the cone. As we know, the parameters, which define a sphere, are radius and origin point in which the radius is constant for the whole shape while in conical shapes the radius is a function of height. In other words, the points in cone surface, which are closer to the vertex, have smaller radius (r) than points near the base of the cone with greater height (h). As we can assume that trees are vertical, it is reasonable to use equation of cone with axis parallel to Z axis. The general equation of a cone is written from Adams (1990) as follows:

$$(x - x_c)^2 + (y - y_c)^2 = (z - z_c)^2 a^2 \tag{1-26}$$

where

- x , y and z are the coordinates of a point on the surface of the cone.
- x_c , y_c and z_c are the coordinates of the cone vertex.
- $a = \tan\alpha = r/h$, the angle α is the opening angle (semi-vertical angle) of the cone.

The aim is to use scanned points (x_i , y_i , z_i) to find $\hat{a}$ as a parameter that defines the best-fit cone with vertex point ($\hat{x}_c$, $\hat{y}_c$, $\hat{z}_c$), which fulfils the least squares condition:

$$\sum(a_i - \hat{a})^2 = \sum \varepsilon_i^2 = min \tag{1-27}$$

where

- $a_i = \sqrt{\frac{(x_i - x_c)^2 + (y_i - y_c)^2}{(z_i - z_c)^2}}$; $i = 1, 2, \dots, n$ (1-28)
  - n is the number of points in the point cloud.

To avoid singularity in Equation (1-28), we do not use points for which $|z_i - z_c| < 0.1m$.

Being able to apply the linear model (1-12), we have to linearize Equation (1-28) by Taylor expansion around approximate coordinates of vertex point (x_{c0} , y_{c0} , z_{c0}):

$$a_i = a_{i0} + \frac{\partial a_i}{\partial x_c}|_{x_{c0}}\, \delta x_c + \frac{\partial a_i}{\partial y_c}|_{y_{c0}} \delta y_c + \frac{\partial a_i}{\partial z_c}|_{z_{c0}} \delta z_c \tag{1-29}$$

where

- $a_{i0} = \sqrt{\frac{(x_i - x_{c0})^2 + (y_i - y_{c0})^2}{(z_i - z_{c0})^2}}$ (1-30)
- δx_c , δy_c and δz_c are improvements to approximate coordinates so that:
  $\hat{x}_c = x_{c0} + \delta x_c$; $\hat{y}_c = y_{c0} + \delta y_c$; $\hat{z}_c = z_{c0} + \delta z_c$

With $a_i = a + \varepsilon_i$, we can rewrite Equation (1-29) in form of Equation (1-10) as:

$$a_{i0} - \varepsilon_i = a - e_i \delta x_c - f_i \delta y_c - g_i \delta z_c \tag{1-31}$$

where

- $e_i = \frac{\partial a_i}{\partial x_c}|_{x_{c0}} = \frac{-(x_1 - x_{c0})}{a_{i0} * (z_1 - z_{c0})^2}$
- $f_i = \frac{\partial a_i}{\partial y_c}|_{y_{c0}} = \frac{-(y_1 - y_{c0})}{a_{i0} * (z_1 - z_{c0})^2}$
- $g_i = \frac{\partial a_i}{\partial z_c}|_{z_{c0}} = -\frac{(x_1 - x_{c0})^2 + (y_1 - y_{c0})^2}{a_{i0} * (z_1 - z_{c0})^3}$

Equation (1-12) becomes:

$$\boldsymbol{L} = \begin{bmatrix} a_{10} \\ \vdots \\ a_{n0} \end{bmatrix} \; ; \; \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix} \; ; \; \boldsymbol{A} = \begin{bmatrix} 1 & -e_1 & -f_1 & -g_1 \\ & \vdots & & \\ 1 & -e_n & -f_n & -g_n \end{bmatrix} \; ; \; \boldsymbol{\delta X} = \begin{bmatrix} a \\ \delta x_c \\ \delta y_c \\ \delta z_c \end{bmatrix} \tag{1-32}$$

The least squares solution is:

$$\boldsymbol{\delta \widehat{X}} = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{L} = [\hat{a} \;\; \delta \hat{x}_c \;\; \delta \hat{y}_c \;\; \delta \hat{z}_c]^T \tag{1-33}$$

The residual matrix ($\boldsymbol{\varepsilon}$) is computed as $\hat{\boldsymbol{\varepsilon}} = \boldsymbol{L} - \boldsymbol{A}\boldsymbol{\delta \widehat{X}}$, which is used for computing standard deviation value for analysing the quality of fitting shape to point cloud. The estimation of errors in unknowns can be performed by Equation (1-16) and Equation (1-17) as:

$$\sigma_{\hat{a}} = \sqrt{\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}}(1,1)}\,, \;\; \sigma_{\delta\hat{x}_c} = \sqrt{\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}}(2,2)}\,, \sigma_{\delta\hat{y}_c} = \sqrt{\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}}(3,3)}\,, \; \sigma_{\delta\hat{z}_c} = \sqrt{\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}}(4,4)} \tag{1-34}$$

where $\boldsymbol{C}_{\delta\hat{X}\delta\hat{X}} = \hat{\sigma}_0{}^2 (\boldsymbol{A}^T \boldsymbol{A})^{-1}$ is a variance matrix of unknowns by dimension of 4*4, and $\hat{\sigma}_0 = \sqrt{\frac{\hat{\boldsymbol{\varepsilon}}^T \hat{\boldsymbol{\varepsilon}}}{n-4}}$

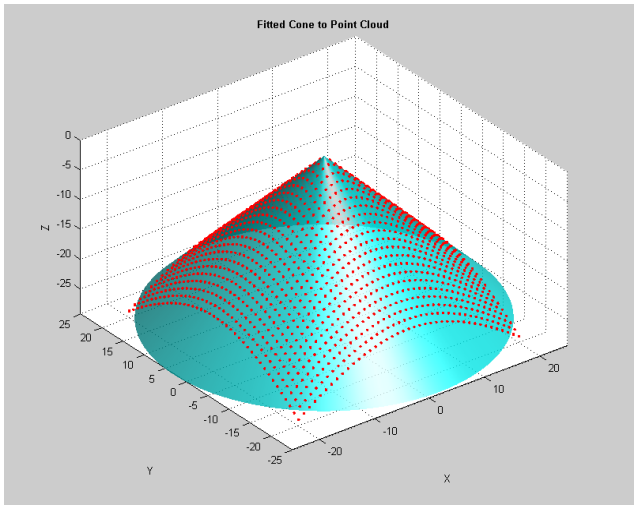


Figure 6: The point cloud for the sample cone is generated with a radius of 25 units, a vertex point of (0, 0, 0) and semi-vertical angle of 45 degrees; as shown in the figure, the algorithm fitted the cone correctly with the same parameters to point cloud.

1.5.3 Fitting a cylinder on a data set

In the previous sections, for fitting sphere and cone, the general equation of each surface was used for the linearization process while in this section; the most general second-degree equation in three variables for quadric surfaces is going to be used for the least squares analysis of best cylinder fit.

$$Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0 \qquad (1\text{-}35)$$

where

$$\begin{aligned} &A = (b^2 + c^2), B = (a^2 + c^2), C = (a^2 + b^2), D = -2ab, E = -2ac, F = -2bc \\ &G = -2(b^2 + c^2)x_0 + 2aby_0 + 2acz_0 \\ &H = 2abx_0 - 2(a^2 + c^2)x0 + 2bcz_0 \\ &I = 2acx_0 + 2bcy_0 - 2(a^2 + b^2)z_0 \\ &J = (b^2 + c^2)x_0^2 + (a^2 + c^2)y_0^2 + (a^2 + b^2)z_0^2 - 2bcy_0z_0 - 2acz_0x_0 - 2abx_0y_0 - R^2 \end{aligned} \qquad (1\text{-}36)$$

A cylinder can be specified by a point (x_o, y_o, z_o) on its axis, a vector (a, b, c) pointing along the axis and its radius (R). Therefore, for fitting a cylinder to a point cloud, these parameters should be computed and finally using least squares method, the distance between any points to the surface of the cylinder being minimised. The workflow of fitting the cylinder procedure is noted as follows[3]:

- Defining the direction vector (a, b, c) that is gained after applying least squares to above general equation.
- Knowing (a, b, c) and using G, H, I coefficients, the initial value for (x_o, y_o, z_o) can be computed.
- From equation of coefficient J, the initial estimation for radius (R) will be calculated.
- As we assumed a cylinder along Z-axis, the data should be transformed by a rotation matrix derived from the direction vector.
- Computing the distances from all points (x_i, y_i, z_i) to the cylinder.
- Solving the least squares system, which minimises the distances between the point cloud and the cylinder.
- Updating the location of the initial origin point
- Repetition of the so-called steps until the system has converged to the limit value.

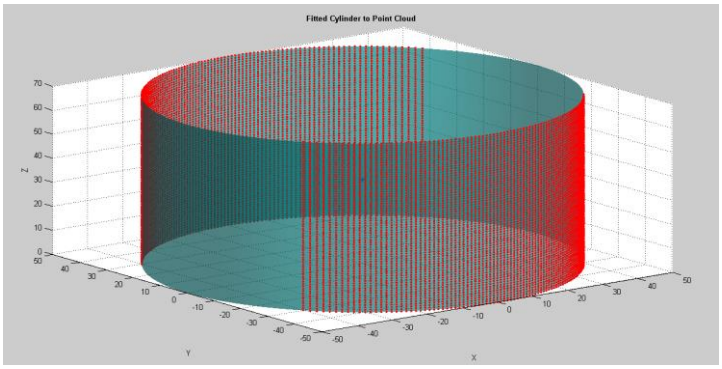


Figure 7: The figure shows the fitted cylinder to generated sample cylinder points with radius of 40 units and origin point in (0, 0, 35); the red points are representing the sample point cloud.

[3] http://www.caves.org/section/commelect/DUSI/openmag/pdf/SphereFitting.pdf [Accessed 4 June 2012].

1.6 Overview of fuzzy logic systems

Basically fuzzy logic has two different meanings, in a narrow sense it is a logical system, which is an extension of a multivalued logic, however, in wider meaning it expresses the fuzzy sets theory, which relates to classes of objects with unsharp boundaries in which membership is a matter of degree (MATLAB documentation). Generally, fuzzy logic analyses the number of input data in some steps and is resulted by output; the mathematical shape of this output may vary based on the type of used fuzzy logic. In the other words, fuzzy logic is a convenient way to map an input space to an output space. Being conceptually easy to understand, offering flexibility, being tolerant to imprecise data, modelling non-linear functions and depending on natural language are the outstanding advantages of fuzzy logic.

The mapping mechanism of input space to output space in fuzzy logic is carried out based on performing a list of *if-then* statements, which are called rules. The input and output terms of rules should be defined carefully to be able to build as efficient a rule as possible. As a summarize of the fuzzy inference concept, *fuzzy inference is a method that interprets the values in the input vector and, based on some set of rules, assigns values to the output vector* (MATLAB documentation).
Fuzzy systems are usually founded by few concepts, which are explained concisely as follows, it should be mentioned that all these concepts are explained with related examples in the MATLAB documentation. MATLAB has a broad and efficient toolbox for creating and performing a fuzzy logic system and some aspects of this are explained in this overview section.

- Fuzzy sets: a set without a crisp, clearly defined boundary.
- Membership Function (MF): is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. In the MATLAB toolbox there are 11 different built in membership function types, which are built based on the piece-wise linear function, the Gaussian distribution function, the sigmoid curve and the quadratic, cubic polynomial curves.
- Logical operations: the input values to the system are connected to each other for making rules using logical operations of *AND* for fuzzy intersection or conjunction, *OR* for fuzzy union or disjunction and *NOT* for fuzzy complement, these operations for two values of A and B are defined classically as $min(A,B)$, $max(A,B)$ and $1-A$ respectively. Figure 8 depicts graphically how they work.

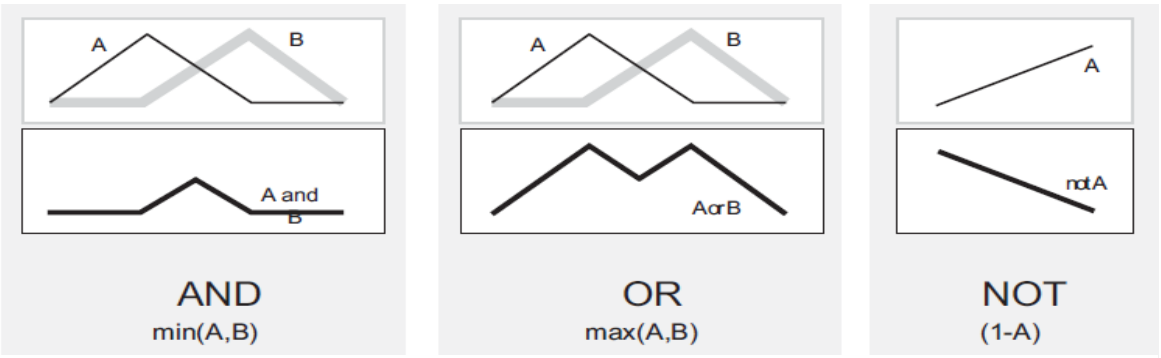


Figure 8: Graphically presents the logical operations in fuzzy logic system; the picture is adapted from MATLAB documentation.

- *If –Then* rule: these statements are used to formulate the conditional statements that comprise fuzzy logic. The if-part of the rule is called the *antecedent* or premise, while the then-part is called the *consequent* or conclusion. For interpreting these statements, first, the antecedent should be evaluated, which involves fuzzifying the input and applying any necessary fuzzy operators. In the second step, the results should be applied to a consequent, which is known as implication step.

Types of fuzzy inference systems (FIS)

There are two different types of fuzzy inference system defined in the MATLAB toolbox, Mamdani and Sugeno. The Mamdani method was proposed by Ebrahim Mamdani in 1975 as a solution to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators (MATLAB documentation). His method is the most commonly used method of fuzzy logic. The Sugeno method was developed by Takagi-Sugeno-Kang in 1985. Both approaches are similar to each other in all steps except the outputs in which the Sugeno output membership functions are either linear or constant and the Mamdani-type inference, expects the output membership functions to be fuzzy sets. It should be noted that these two types are able to convert to one another in MATLAB, by a single command. The fuzzy inference process consists of five steps and as mentioned are the same for both methods.

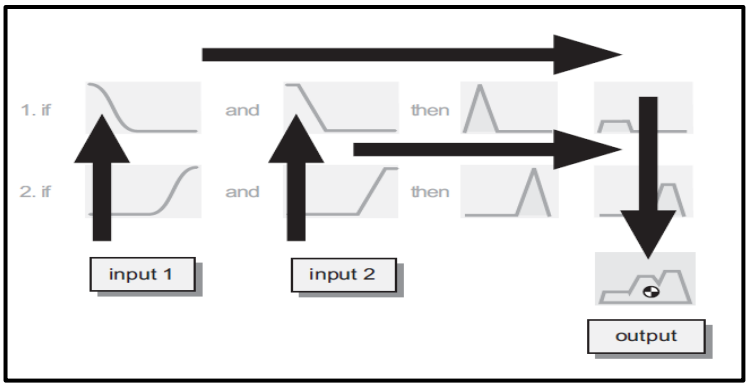
1. Fuzzification of input variables: The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions.
2. Application of the fuzzy operator (*AND* or *OR*) in the antecedent: as explained before, in this step, logical operators are used to derive a value as a result based on mathematical operations.
3. Implication from the antecedent to the consequent: The implication function modifies the fuzzy set to the degree specified by the antecedent. The most common ways to modify the output fuzzy set are truncation using the min (minimum) function or scaling using the $prod$ (product) function.
4. Aggregation of the consequents across the rules: In this step, the fuzzy sets, which are representing the outputs of each rule are combined to a single fuzzy set as a result (only in Mamdani type).
5. Defuzzification: the input of this step is an aggregated fuzzy set from the previous step and the output is a single number. This step is different for Mamdani and Sugeno types, as in the Mamdani-type the input fuzzy set for defuzzification is an aggregated membership function, so the output value would be computed by one of the built-in functions such as the centroid of the area under the curve (the most common function for defuzzification in Mamdani) while in Sugeno-type the final value is computed based on a value of each rule and assigned weight as this formula:

$$Final\ Output = \frac{\sum_1^N W_i Z_i}{\sum_1^N W_i} \tag{1-37}$$

where N is the number of rules, Z_i is the value of each rule and W_i is the weight value.

All these steps are graphically explained in Figure 9 for Mamdani type.

Figure 9: The diagram illustrates the general process (steps 1 to 5) of the Mamdani type; the arrows show the sequence of process. The number of rows is denoted by the number of rules and as can be seen from the picture the output is computed through a centroid function. The picture is adapted from MATLAB documentation.



1.6.1 Building the Fuzzy Inference System (FIS) in MATLAB

In the MATLAB toolbox for fuzzy logic, there is a capability to develop a FIS, either using GUI interface or writing codes, it is also possible to create a system as an interface and edit the source codes or evaluate the final outputs. Building and interpreting such a system is feasible by the following five steps.

1. FIS editor: specifying the number of input and output variables with their corresponding names.
2. Membership Function (MF) editor: defining the shapes of all the membership functions associated with each variable.
3. Rule editor: forming new rules or editing the existing rules, which are defining the behaviour of the system.
4. Rule viewer: in this step, the system gives us an overview of what has been performed from the beginning, all the rules, the result of the aggregation step and also the defuzzified value can be seen in that window or can be extracted by using built in codes.
5. Surface viewer: the dependency of one output on one or two inputs is plotted in a window.

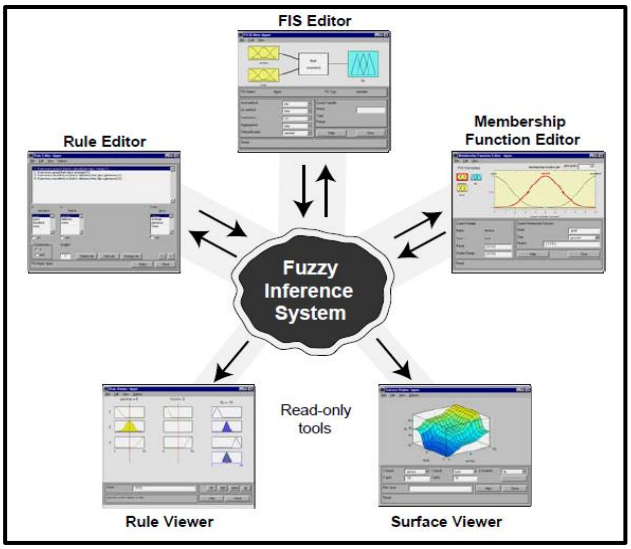


Figure 10: Fuzzy Inference System (FIS) in MATLAB

2 Materials and Methods

In this study, the author writes all algorithms for detecting trees and identifying their corresponding species in MATLAB (a high-level language and interactive environment for numerical computation, visualization, and programming developed by MathWorks®). Manipulation of data is carried out by Cyclone (3D Point Cloud Processing Software developed by Leica Geosystems) and DPM (developed by Visimind AB for LiDAR data processing) software. In the validation part, Arc GIS 10.0 and Quantum GIS are used for doing some analysis and converting shape files to .kmz files.

2.1 Study area and data

The data for the thesis work was provided by the Visimind Company, which is a company based in Sweden and is dealing with remote sensing (laser scanning), photogrammetry and GIS. From their last scanning project, two forestial areas near Stockholm and Borås (in south west of Sweden) are used. The first part of thesis is focused on tree detection. The test area shown in Figure 11 is chosen from an area located in the east side of Stockholm. The selected area covers 2100 m^2 of forest. Further analyses were performed on the other test sites with a total area 3000 m^2, which can be seen in left picture of same figure.

Figure 11: The left picture shows three different test areas represented by tree symbols around Stockholm and the right picture shows the zoomed image of test area, which is located in the east side of Stockholm near to Värmdö town. The inner picture illustrates the point cloud of area.

In both test areas, either east or south west of Sweden, more than 70 percent of tree species belongs to coniferous trees like pine and spruce and the rest are broad-leaf and deciduous trees like birch, oak and aspen. In the first part, we are not interested in tree types and the goal is determining the number of trees to be able to perform clustering methods.

In the second part of the study, the data was used for tree species categorization. For this purpose, the species of several trees in the reference site were distinguished by field measurements using GPS technique (Figure 12*a*) and for further accuracy, all measured points were checked with their related images taken at the time of scan. DPM has such capability to show laser data and images of scanned area in different windows. As shown in Figure 13, the right picture represents one of the scanned areas

with corresponding tree species names for distinguished trees. The reference sites are located near to Borås and Lerum cities, south west of Sweden. Totally 280 trees were distinguished in sites and used as reference trees. Despite 280 known-species trees, 123 trees were used in this study because of dense forests in which trees grow very close to each other and make it difficult to export laser points of single trees among others. In Figure 14, sample exported tree points from each species was shown. Finally, 33 birch, 45 pine, 40 spruce and 5 oak trees were exported for performing the second part of this study.

(a) *(b)*

Figure 12: Picture (*b*) shows spruce trees in the test area, which was taken during field measurement and picture (*a*) was taken during measuring by GPS.

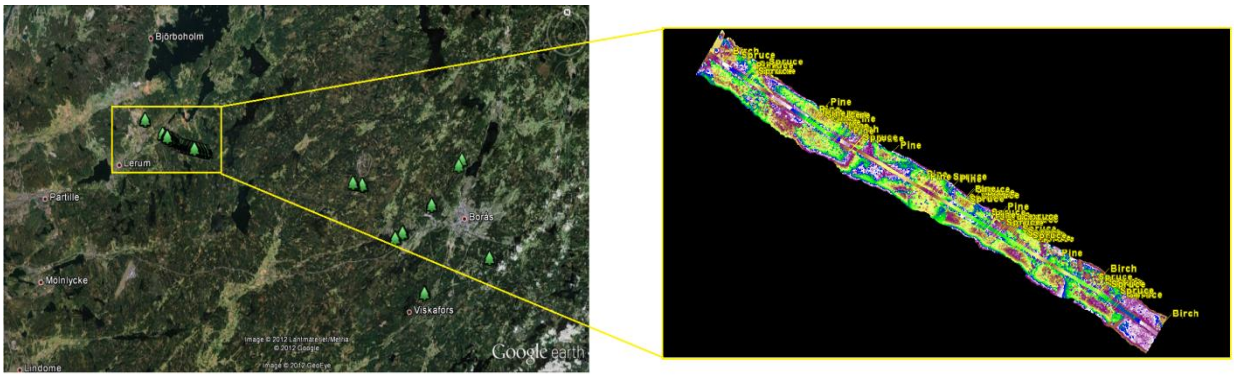


Figure 13: Test areas close to Borås and Lerum cities in south west of Sweden are shown in the left picture, the areas with known-tree species are shown by green tree symbols and the right picture shows the point cloud of the area with coordinates of distinguished trees, which are shown inside the point cloud as label of trees.

Figure 14: Exported laser points for different tree species, which exist in the test areas.

As mentioned before, this thesis is carried out based on two sets of laser scanning data, which belong to an area near Stockholm called *Mellan data* and data acquired from west part of Sweden called *Väst data* in the Visimind database. The *Mellan* data was acquired by using a RIEGL VQ-480 scanner during September 2011 and *Väst* data was obtained using the RIEGL VQ-380 scanner in early October of the same year. So the laser data can be considered as leaf-on data. The specification of both scanners with corresponding computed point density of data is written in Table 1. These scanners are developed by the RIEGL Company under *V-Line* series of laser scanning engines in 2010; their mechanism of working is based on a fast rotating multi-facet polygonal mirror, which provides fully linear, unidirectional and parallel scan lines (RIEGL website[4]).

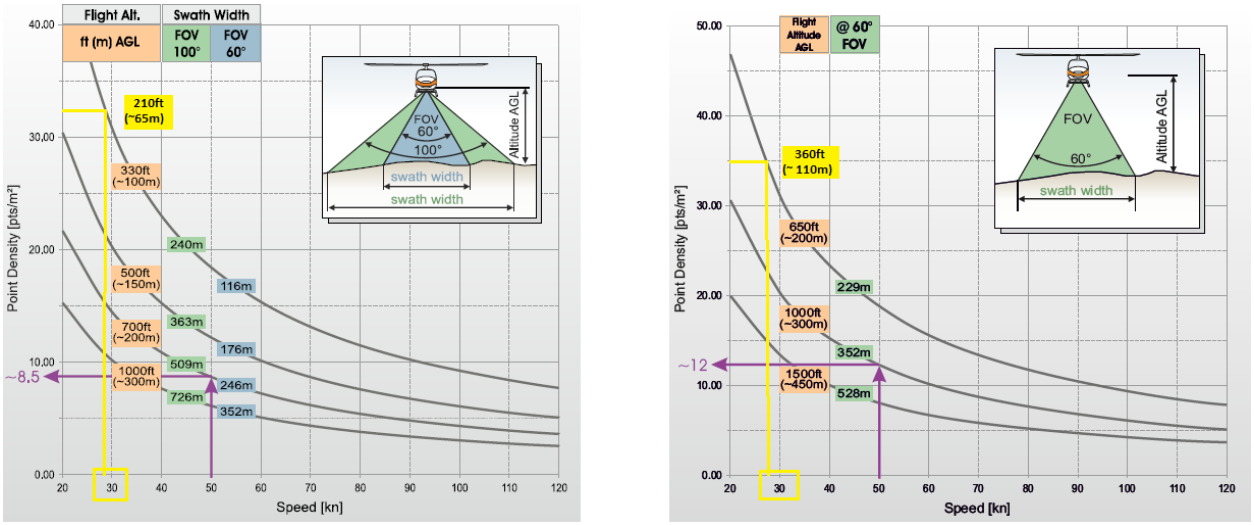
The exported information from the laser data include three coordinates of points (X, Y, Z), RGB values and intensity values, also the coordinates were projected to SWEREF 99 (Swedish Reference Frame 1999) coordinate system.

Table 1: The specification of RIEGL scanners used for producing laser data from test areas is shown.

Type of Scanner	Laser Pulse Repetition Rate (PRR)	Field Of View (FOV)	Swath Width	Flight Altitude	Helicopter Speed	Point Density
RIEGL VQ-380	200 *kHz*	100˚	160 *m*	~65 *m*	~55 *km/h*	35 *pts/m²*
RIEGL VQ-480	200 *kHz*	60˚	130 *m*	~110 *m*	~55 *km/h*	35 *pts/m²*

For filling the specification of Table 1, the information for helicopter speed was not available in the flight documents of the Visimind Company, so using the published specification of scanners in RIEGL website and using other known information, the speed is derived from the graph shown in Figure 15.

[4] www.riegl.com



a) VQ 380 *b) VQ 480*

Figure 15: The specification of VQ-380 and VQ-480 scanners published by RIEGL and derived speed amount using the relation between point density, speed and flight height for so-called scanners with 200 kHz laser frequency. The yellow lines and boxes were drawn by the author.

2.2 Detecting the number of trees

As explained in the previous sections, there are different methods to derive the number of existing trees in forestial or urban areas and mostly are being counted based on aerial images or the interpolated surface of canopy. In the algorithm, which is written for this thesis to perform tree detecting, the study area is considered as a matrix so the X and Y coordinate of an area is represented by the rows and the columns of a matrix; here the elevation element of area is ignored. In the second step the area based on its shape and how it extends (along X direction or Y), the matrix would divide to different strips in which the width of the strips has direct relation with the average crown-diameter of the existing trees in that area. In other words, the area is assumed to be made of several strips in which trees are located along one line beside each other and this can be described as a profile of that strip. The idea came from the so called DPM software that has such a capability to show the selected scanned area as a cross section, which makes it easier for the viewer to determine the number of trees; this is shown in Figure 16. Nowadays the other new developed software, which is concerned about laser scanning data, has or adds a feature to perform cross section on point clouds. The cross section feature can be applied in either urban or forestial areas for detecting trees; for instance the number of trees along one street inside the city can be considered as an object for applying this method; a strip with a width of tree crown-diameters and the length of selected part of street would be a representer of that area. In Figure 17, the selected forestial area is shown, which is divided to six strips and the profile line is drawn based on the outer surface of the tree row. For this area, the width of the strips is computed based on the crown-diameter of the sample measured trees of spruce, pine and birch in that area, which is six meters and cross sections are generated along the X direction. It should be mentioned that the X and Y axes represent the length and width of the strips, respectively and the Z axis shows the height of trees.

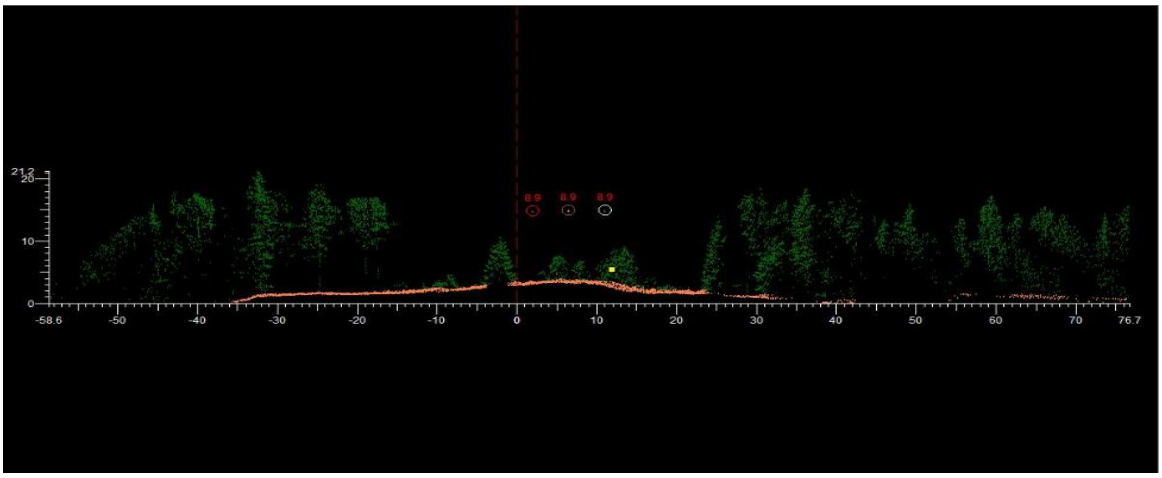


Figure 16: Cross section for selected part of laser data created by DPM.

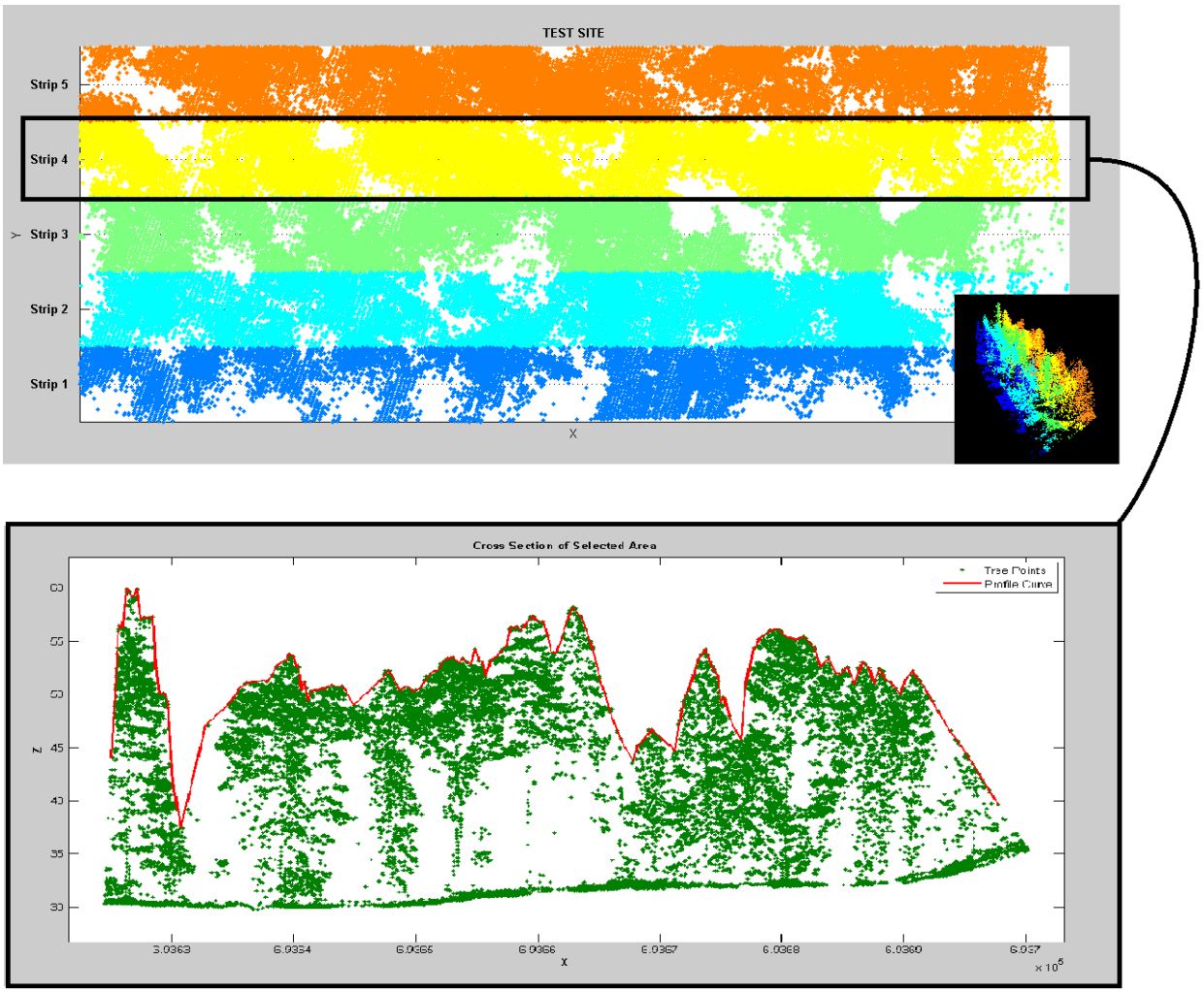


Figure 17: Divided area to five strips and profile curve is drawn on cross section of strip 4.

Local Maxima and Local Minima

The main objective of dividing the area into strips is to be able to find the number of trees in each strip because for performing clustering analyses, the number of clusters should be determined in advance. As shown in Figure 17 the drawn profile curve on trees is including several maximum points that can be interpreted as top points of trees, so the maximum points would lead to top-coordinates of any detected tree in its corresponding strip. The written algorithm for this purpose, firstly finds the outer points of laser data, which are the peak points of all existing points in short intervals of X coordinates. In the second step, each exported outer point is being compared in aspect of Z value with its back and front points to being classified as local maxima, local minima or none of them. The pseudo code is as follows:

```
a) if peak(i) < peak(i-1) and peak(i) < peak(i+1) then peak(i) is
   Local Minimum
b) else if peak(i) > peak(i-1) and peak(i) > peak(i+1) then peak(i)
   is Local Maximum
c) else peak(i) is an ordinary outer point
```

Modified Local Maxima (Seeds)

Usually trees do not have fixed and defined behaviours in their growth and they are not following any specific rules. The developed algorithms for tree detection are tested on several areas and it is noticed that the local maxima point clouds do not define accurately the top points of the trees. The spread out branches of some trees lead to more than one local maximum for one tree. Therefore, two filtering steps were added to the algorithm to decrease the influence of redundant local maxima points in clustering.

- In the first approach, the distances between pairs of local minima and maxima points are computed in X-Z plane, in other words, any local maxima point is being compared with two local minima points before and after that point. If those distances exceed two meters and the point belongs to previously computed local maxima group, that point would be selected as Modified Local Maxima, which is called Seed for clustering. The threshold distance is determined empirically based on the trial and error method on several test strips and finally the optimum value of two meters was obtained. Hence, it is probable to get different threshold value in other type of forests.
- As explained before, when we involve the large area, we divide it into several strips, but a problem would occur if a tree is located in the boundary of two strips. In this case, two strips would have a partial point cloud of one tree and in each strip, a local maxima point would be allocated to that partial point cloud; for removing such redundant points, the second approach was developed for filtering based on all possible distances between local maxima points in X-Y plane. The chosen threshold is 2.0 meters, which is determined based on narrowest crown diameter of reference trees. Age and type of trees as well as forest density are major factors to determine such a threshold value for filtering data. The input of this method is the modified

local maxima of all strips from previous filtering step. Figure 18 and Figure 19 show the result of these two filtering algorithms for one strip. The pseudo code for X-Y filtering and exclusion of redundant points is as follows:

a) Set i = 1
b) Compute the Euclidean distance D_{ij} between any local maxima point P_i and all other remaining points
c) Define threshold distance D_{th}
d) If $D_{ij} > D_{th}$ select point P_i as a local maxima point and assign to *seed* matrix, else reject P_i
e) Increment i = i + 1
f) Continue the loop until all distances from all points are being compared with threshold value.

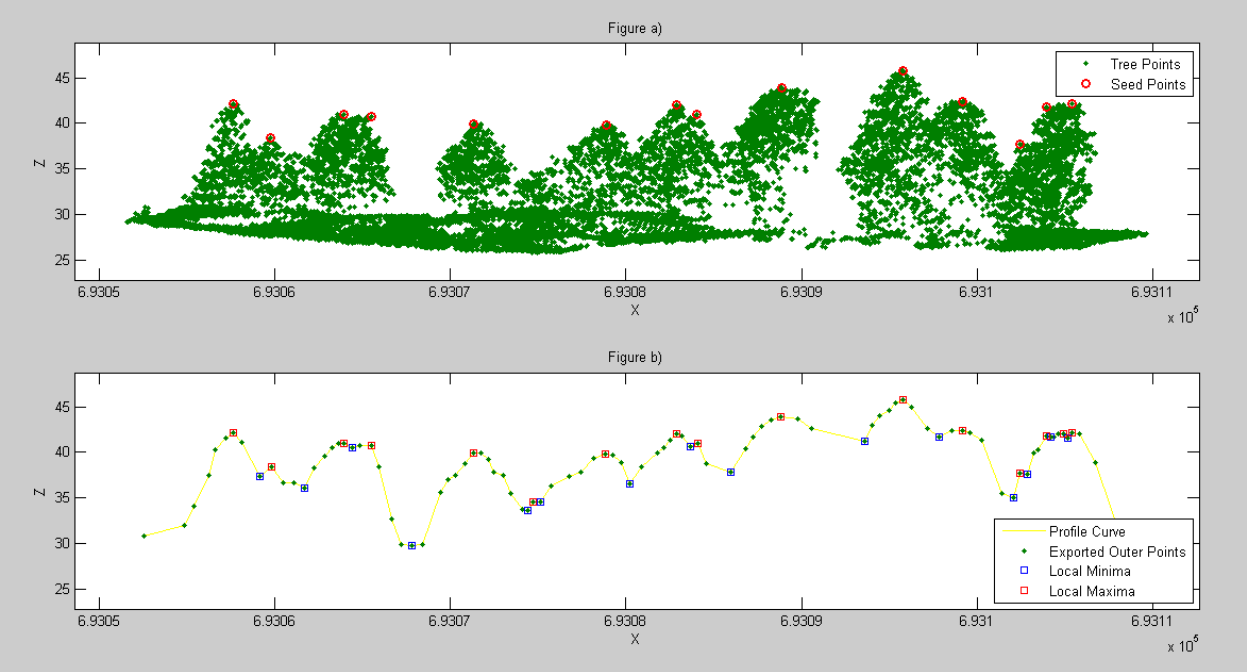
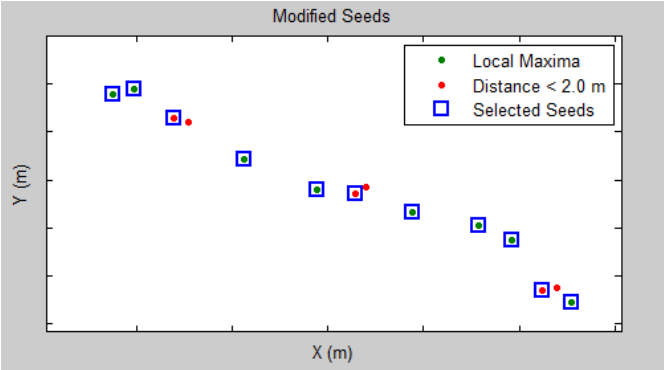


Figure 18: Picture b) shows the local minima and maxima points in blue and red squares, respectively. These points are derived from the green outer points, which are exported from the point cloud of strip, shown in the upper picture (a). In picture a) the result of X-Z filter are shown as seed points by red circles, which are representing the top points of each tree and can be counted as the number of trees in each strip.

Figure 19: Modified Seeds exported from the second filtering (X-Y) step for the same area of the previous Figure. The local maxima points having less than 2 meters distance to each other are plotted in red colour. One of those red points located beside each other, is chosen as the modified seed point. These points also delineate the number of trees in the investigated area.



2.3 Clustering algorithms

2.3.1 Unsupervised K-means clustering

As outlined in the introduction part, the K-means is an iterative partitioning-based method that requires the user-specified target number of clusters (K). The algorithm generates K data points representing cluster centers or centroids. An iterative partitioning loop is then executed to carry out the clustering. In first study, the Unsupervised K-means approach is applied to data set, in this method the number of random initial point is kept equal to the number of external seed points to get the same number of clusters as other methods. The initial points are created randomly by algorithm.

2.3.2 Supervised K-means clustering

This approach is called supervised because of using external seed points as initializing points for the algorithm. The modified local maxima points, which were detected and prepared in the previous section are going to be used as external seed points for this method. The K-means approach uses those points as initial points for starting the iterative procedure of clustering.

Using clustering methods in forestial areas for clustering the whole height of individual trees is not completely possible. In dense forests, the trees are so close to each other that in the middle section of the trees, it is difficult to draw a boundary for tree crowns. Applying clustering methods for separating the whole tree would not be efficient and usually some points from neighbouring trees are clustered as one tree. Therefore, we decided to test methods on the top 60 percent of height for less dense forest areas and top 40 percent of height for dense areas. It should be mentioned that for the area, where there are gaps between trees as urban areas, the algorithm could be run for whole height. In addition, it is noted that K-means works well when a data set has compact or isolated clusters (Mao & Jain, 1996). The pseudo code of Supervised K-means applied on a 3-dimendion data set is given below (Gupta, 2010):

a) Set $i = 1$
b) Define external seed points as a set of K-means $C_1(i)$, $C_2(i)$,…, $C_k(i)$
c) For each point P_i compute the distance D (between P_i and $C_k(i)$, $i = 1{:}k$) then assign P_i to cluster C_j based on nearest Euclidean distance
d) Increment $i = i + 1$ and update C_j to get new set of $C_k(i)$
e) Repeat steps a) to d) until $C_k(i) = C_k(i + 1)$ for all k

2.3.3 Hierarchical clustering

The principle of this clustering method is completely explained in section 1.2.1. Here it should be noted that the weighted average distance (WPGMA) method used as a component of the linkage function for applying hierarchical clustering on a data set. There is no input as seed points for this approach but it

works based on the investigated similarities between objects; for getting a result, which can be compared with the results of K-means, the number of desired clusters is given as input to this algorithm, so the dendrogram has been cut based on the given cluster number. It is also possible to get clusters without defining number of groups, by using inconsistency values.

2.4 Validation procedures for tree detection and clustering

For validating the result of this part of the thesis work, which consists of detecting tree tops and clustering using three methods, different types of accuracy assessment are used.

1. For expressing accuracy of tree detection in percent, two types of accuracy classes are commonly used. The first type is *user's accuracy,* which is equal to sum of correctly detected trees divided by all detected trees and second type comes from dividing sum of correctly detected trees by sum of reference tree tops, which is called *producer's accuracy*. It shows the probability that the reference trees are detected correctly (Gupta, 2010). In this study it was not possible to validate the results by reference trees due to lack of field measurements, instead the first type of accuracy class was used, and the detected trees were compared visually by original LiDAR data in Cyclone and DPM.
2. In the second step, the obtained results were compared with the results from DPM. The comparison was performed in ArcGIS. The accuracy assessment was carried out in two levels explained in detail in section 3.1.2. In the first level of assessment, point shapefiles, which are visually overlaid, counted as exactly matched trees. In the second level, a buffer of 2.5 meters was created for DPM points and detected tree points from the thesis method located in that zone, were counted as nearly exactly detected trees. A brief explanation of how the tree detection tool is working in DPM is written below:

Tree detection tool in DPM

By selecting an area of laser points in the tree detection tool of DPM, it creates a normalised digital surface model (nDSM) of canopy of trees. Using neighbour analysis of each pixel, it finds the pixel with higher value as centre point, and starts watershed method to find all pixels, which are less than that pixel and assigns them to one tree, this procedure is performed iteratively for whole nDSM and detected trees are delineated by cylinders. The height of trees computed from the subtraction of digital surface model (DSM) and the digital terrain model (DTM) for the central pixel (Figure 20).

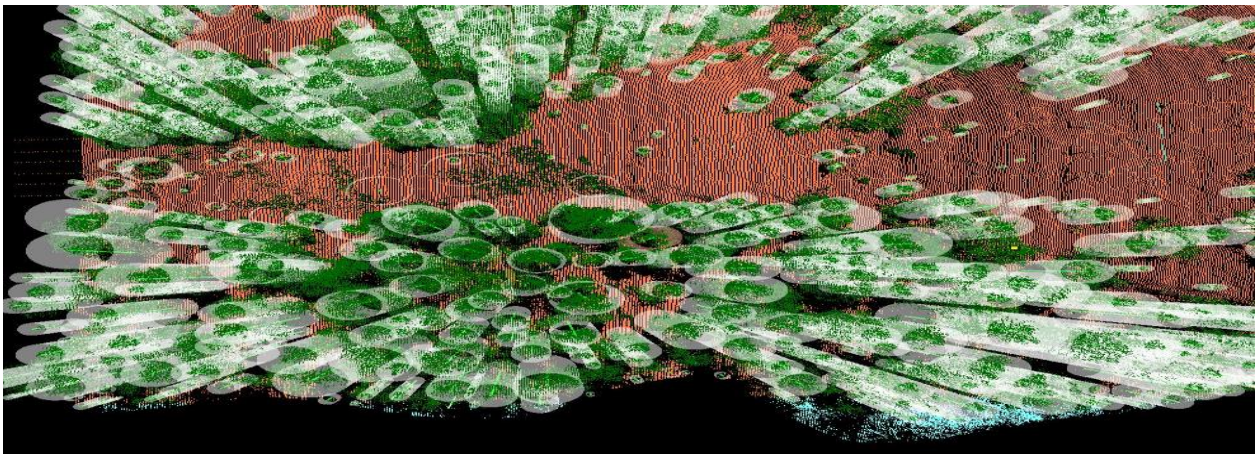


Figure 20: Tree detection by DPM software.

3. Precision analysis was performed for data sets in which three type of clustering methods were applied to find out the best approach for clustering based on mean silhouette and mean standard deviation values of all clusters. In addition, the user's accuracy can be computed from dividing the sum of correctly and completely clustered trees by the number of clusters, which was carried out visually.

2.5 Tree species classification

The second part of this study involves tree species identification based on an algorithm developed by the author. After detecting the trees, it is required to classify the types of trees. As noted before, for this purpose different authors did use different approaches; the most usual technique is using spectral and aerial images for identifying tree species. Here in this study, we use only laser point data for both, tree detection and species classification. This data is usually obtained by scanning the determined area by a helicopter or aircraft in any time of the year and depending on the flying season, two different data sets (leaf-on or leaf-off) are obtained. Although it is ideal to get benefit of both types of data, usually one type is accessible for any area. In this study, the leaf-on data is used and as our studied sites are combined by both deciduous and coniferous trees, so leaf-on data would provide more detailed information of deciduous trees. In spite of other studies, the geometric shape of trees is playing a significant role in our algorithm. As mentioned by (Ko et al., 2009), LiDAR point clouds do not only provide us information about the surface shape but also some useful information can be derived from the inside of the crown that is worth to be considered.

As mentioned in section (2.1), three species of trees (birch, spruce and pine) were extracted as reference trees for testing our algorithms and finally the accuracy assessment was performed based on those trees. Several algorithms were developed for performing analysis on both, outer surface of trees and inner point clouds. Finally, a fuzzy logic system is used for aggregating the different methods for getting a unique result. It should be mentioned that we performed clustering methods on the upper half of the point clouds and as we will apply the obtained fuzzy model on clustered tree points in the final

step for identifying species, so the same portion of individual reference trees is involved for defining optimum fuzzy model.

2.5.1 Fitted shape analysis

For classifying tree species, their shape characteristics are analysed in this section by fitting a 3D geometric model to the point cloud. These shapes are fitted based on the least squares method and minimising the Euclidean distance of all points to the shape surface as explained in section (1.5). Three geometrical shapes of cone, sphere and cylinder are fitted and their corresponding standard error, which represents how precise the fitting is, saved for further computations. These three shapes were chosen because deciduous trees are associated with spherical crowns and coniferous trees are associated with conical and cylindrical shapes (Horn, 1971). Cylinder shapes can present *columnar* tree types such as European hornbeam, Lombardy poplar, quaking aspen, etc.; trees with wide base and narrower top that are known as pyramidal species can be presented by cone shapes like American beech, American holly, bald cypress, spruce, cucumber magnolia, fir, linden and sweet gum. *Globe* type trees with rounded shape canopies such as American hornbeam, American yellowwood, oak, maple, flowering dogwood, hackberry and redbud are good options for being tested with sphere modelling. It should be denoted that this algorithm, which fits three shapes at the same time to tree points, is applied to all of exported trees individually. Figure 21, shows how it works for one sample tree. In our reference tree set, there are five oak trees, which were extracted from the study site, for species analysis. As we could not find enough oak trees for involving in the species classification process, those trees were ignored. Despite ignoring those species, for proving efficiency of the written algorithm in classifying deciduous-coniferous trees, those five oak trees were used for comparing spherical and cylindrical fitting with corresponding standard error values. As expected and can be seen in Figure 22, the spherical model works better for recognizing these deciduous trees.

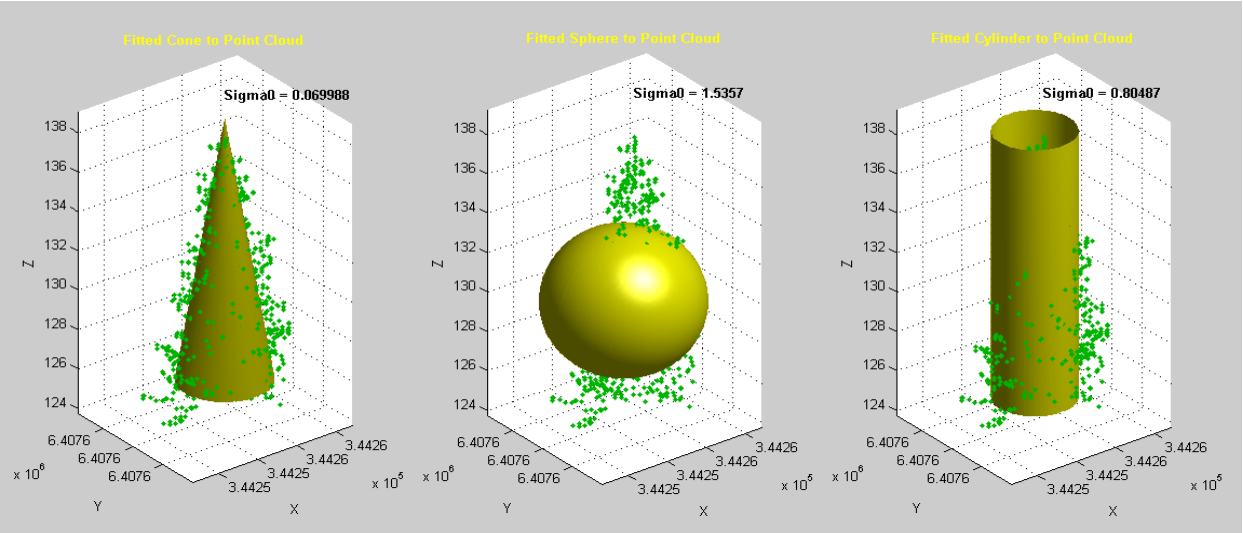


Figure 21: Three geometrical shapes are fitted to tree point cloud and the standard error for each fitting shape is printed above each plot.

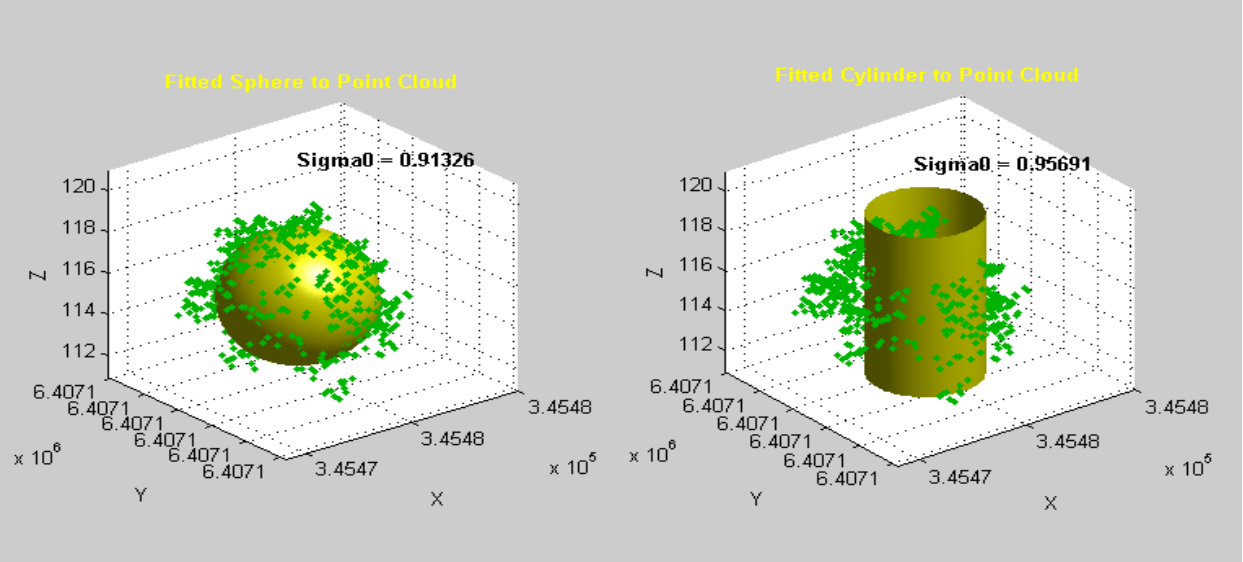


Figure 22: The algorithm was applied to the oak tree and it is clear from the standard error values, which are mentioned as *Sigma0*, that the spherical shape is the better estimation for oak tree due to its spherical crown.

2.5.2 Hull ratio calculation

Convex hull is the other factor for deriving more information from geometrical shape of trees. In mathematics, an object is called convex if for every pair of points within the object, every point on the straight line segment that joins them, is also within the object. Formally, the convex hull of a set X of points in Euclidean space is the smallest convex set that contains X. Implementing convex hull on individual tree points, yields a volume of tree crown and combination of triangular patches, which made the hull. Defining the hull ratio concept, a numerical estimation of the tree crown shapes can be achieved as follows:

$$HullRatio = \frac{A_{hull}}{V_{hull}} * \frac{h_{crown}}{2} \quad (2\text{-}1)$$

where

- V_{hull} is the volume of the convex hull of the tree crown.
- A_{hull} is the surface area of the convex hull, which is computed as the sum of all triangular facets comprising the hull.
- h_{crown} is the height of tree crown, which is computed as the difference of maximum and minimum height of tree (Ko et al., 2009).

This number is equal to 3 for a presumed sphere and by deforming the shape this number would also change to a larger value. The hull ratio is used in this study for investigating the sphere nature of trees, the closer to 3, the more spherical the tree shape is. This would be a good factor to distinguish deciduous trees such as oak, maple and hornbeam. Figure 23 shows a convex hull of one sample tree.

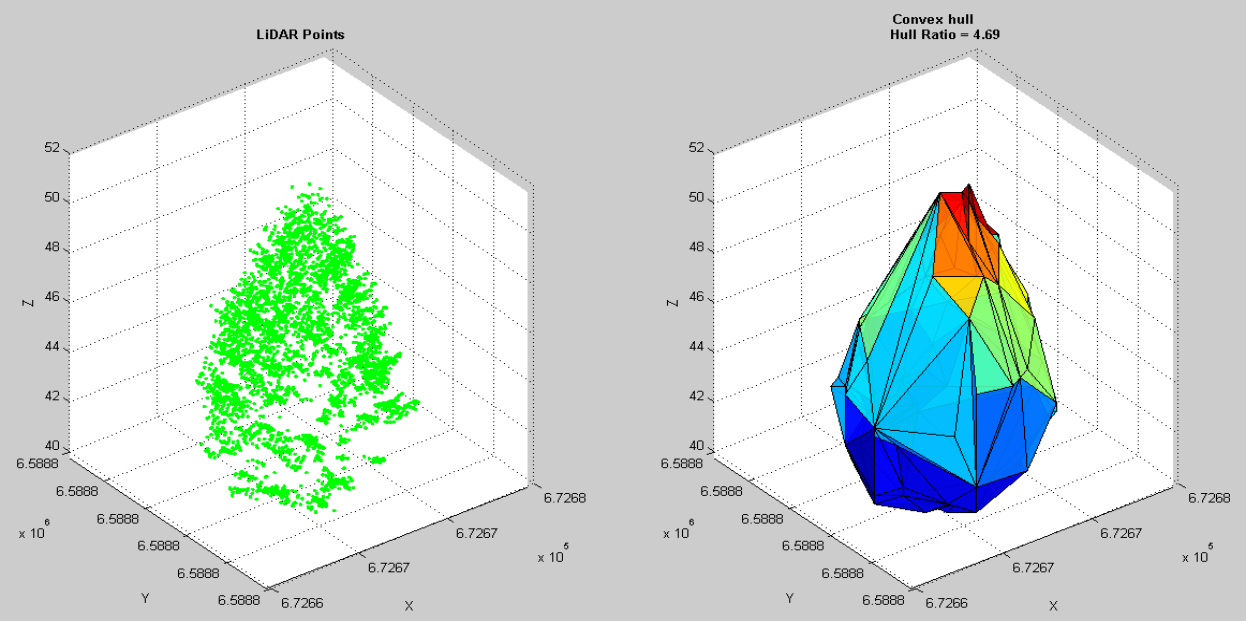


Figure 23: A convex hull, which is drawn for a spruce crown, the printed hull ratio value, shows how much this crown shape differs from a completely spherical shape. The left picture shows the LiDAR point data of a spruce crown.

2.5.3 Density calculation

As already explained, LiDAR data may provide us with useful information about inner structure of trees and as we know, laser beams could not penetrate so deep in the trees with very dense leaves on their top part so the idea has been formed to test one of common concepts of data mining on trees. Point Density in data mining is a general concept being performed on data based on its different types. Grid density, Sphere density and Kernel density are different sub sections of point density[5]. Grid density is the equivalent of the traditional number density in which Space is split into cells, and the density in each cell is simply the number of reference points contained in it. For sphere density, the number of points in a sphere around the target point is counted. A kernel is a weighting function whose value is 1 at the centre and which falls off to zero the further you get away from its centre. To calculate the kernel density, a kernel is placed at each point in the point set. The density of a point is then defined as the sum of all kernel functions at that point. For this study, Sphere density was used for finding out how dense the different species of trees are, for instance it is expected to get a lower density value for birch trees due to their sparse branches versus dense leaves of pine trees, which may lead larger density value. Generally, in sphere density the number of points within a sphere around target point is being counted as point density. The work flow of computing point densities in our developed algorithm is as follows:

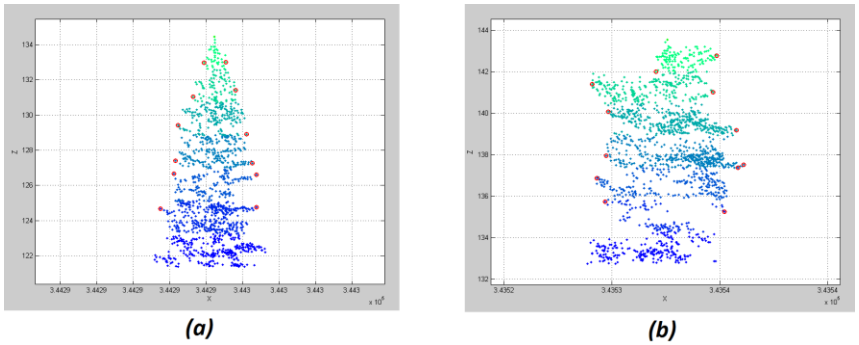
- The Euclidean distances between all points of the tree crown are computed.

[5] http://www.florianbrucker.de/index.php?p=pointclouddensities

- The threshold number as radius of sphere, which defines the boundary of neighbourhood for each point, should be defined. For this study, a radius of 20 *cm* was empirically chosen. It should be mentioned that this number would vary in analysing different data sets.
- Points with distances equal or less than 20 *cm* according to other points are selected.
- And finally $Density\ Ratio = \frac{number\ of\ selected\ points}{number\ of\ all\ points}$

2.5.4 Slope changes

The last factor, which was developed for species identification, analyses the superficial shape of tree in the projected X-Z plane. When the trees are projected into 2D planes, their geometrical shape can be described mathematically according to the slope changes of the outer surface shape. For instance, the outer surface of spruce is smooth and depicts sharp slopes in both sides while pine trees have branches spreading out, which cause more dents and gentle slopes. As can be seen in Figure 24, which shows a spruce and a pine tree, the algorithm divided the trees to seven sections and found the outer points in each section. Then the gradient of the sections is computed using the extracted points . In the next step, the sign of the gradients are analysed to find how many times the slopes are changed. More irregular shapes of the tree yields a greater number of different signs. It is expected for the pine trees to get more irregularities than spruce. In addition, the average slope of the tree in both sides can be computed for further use.



(a) (b)

Figure 24: Image (*a*) shows one spruce tree with smooth outer surface versus image (*b*) that illustrates a pine tree with more dents and irregular shape. The different colours of the point cloud show so-called seven sections and red circles are representing outer points in each

2.6 Fuzzy logic based tree species classification

As explained in the previous sections, for identifying tree species from LiDAR data, four different methods were used but due to the lack of a clear and distinct boundary in the output result of each method, we could not be able to decide exactly about the species of trees. In other words, any single method was not completely sufficient for determining the tree types because there is always a common range of outputs between two or three spruce, pine and birch trees. Hence, it is crystal clear that using only an individual approach for species classification would lead to low accuracy. The simple solution was aggregating the outputs of all methods, but the system, which could be efficient for this purpose, was a matter of decision. It has been decided to use a fuzzy logic system due to its popularity and efficiency in gathering fuzzy sets without clearly defined boundaries. The Fuzzy logic system in MATLAB

is supported by GUI interface so rather than programing for inserting data, one can use that interface facility.

The workflow of building fuzzy inference system for this study is explained in several steps as follows:

First; the number of input and output variables should be determined as well as the FIS method for analysing the data. Our goal is to determine the species of trees such as birch, spruce or pine, based on the concepts of fuzzy systems. Sugeno type fuzzy inference fulfils our requirements as it is used for constant output variables. Four methods, which were already explained, are defined as input variables and it should be mentioned that the 3D shape fitting method, comprises three different approaches, so totally six variables are defined as shown in Figure 25. Three constant variables are also defined as output in the FIS editor window.

Figure 25: Six input variables (Point Density, Hull Ratio, Slope Changes, Fitted Cone, Fitted Cylinder and Fitted Sphere) and three output variables (pine, spruce and birch) are defined in the FIS editor window.

Second; the membership functions should be defined for each variable; birch, spruce and pine are three MFs for each variable in this fuzzy system. These MFs are usually defined based on the built-in function types. In this study, two basic and simple function types were used, *trimf* and *trapmf*. These types were chosen based on the output histogram shape of each method. For instance, the result of the density calculation method is shown by the histogram in Figure 26, which defines the type of MFs. The triangular-shaped function (*trimf*) defines a triangular curve with two parameters for locating *feet* part and one parameter for defining the *peak* point of triangle. The equation of this function can be written mathematically as equation 2-2, where x is a vector data, a, c are *feet* parameters, and b defines the *peak* parameter:

$$f(x;a,b,c) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases} \quad (2\text{-}2)$$

The trapezoidal-shaped function (*trapmf*) is also used for some other data with a histogram-shape of trapezoidal and is defined by a, d as *feet* parameters, and b, c as parameters locating *shoulders* as presented by the following equation:

$$f(x;a,b,c,d) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \\ 0, & d \le x \end{cases} \quad (2\text{-}3)$$

The range of input space for each variable is derived from its corresponding scatter plot, which was drawn separately for each method as shown in Figure 26. Each point in the input space is mapped to a membership value (between 0 and 1) by the membership function curve, in other words, the MF curve defines the probability of any point in the input space.

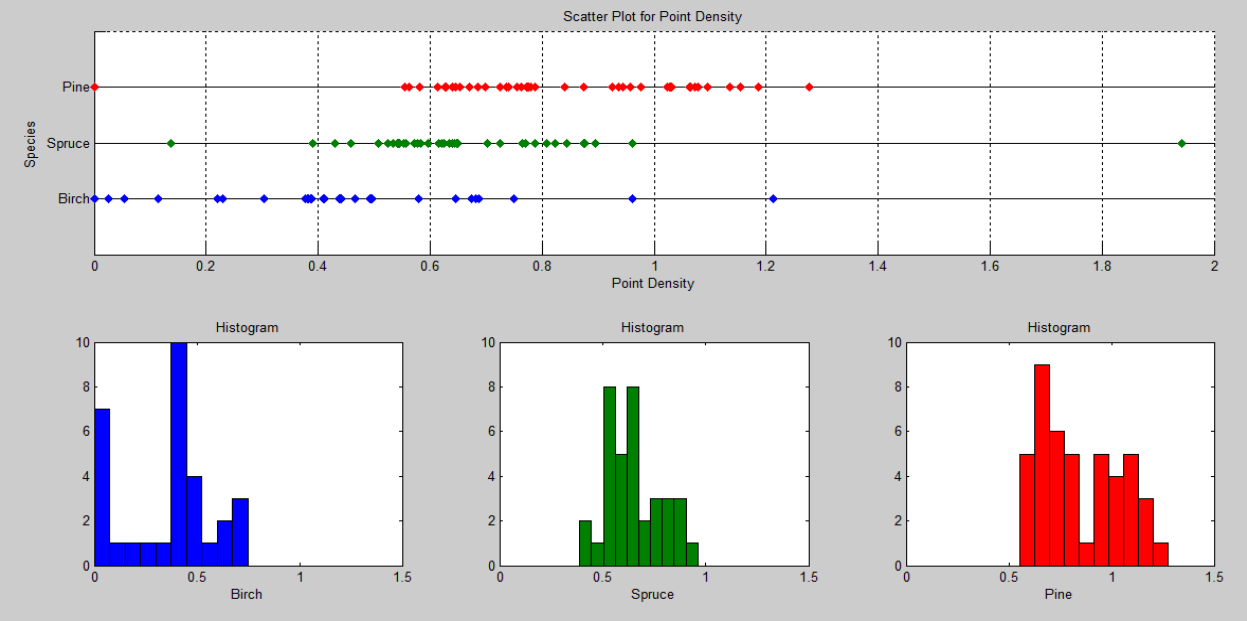


Figure 26: The scatter plot and histogram are drawn for each variable (in this picture for Point Density) to model the shape of the output data from each method for defining MF curves and input space range. It should be mentioned that histograms are created after removing outliers.

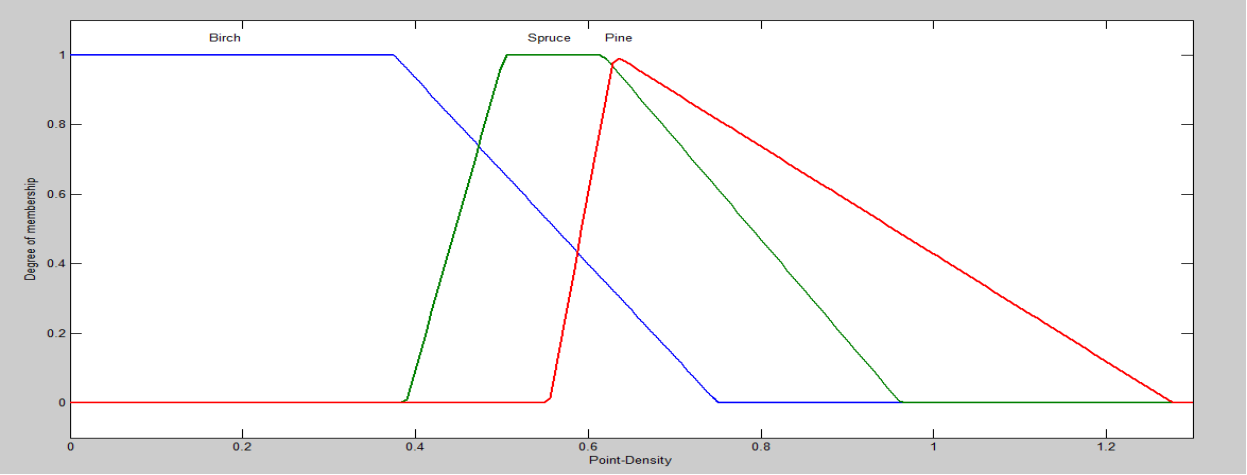


Figure 27: The MFs are drawn for point density variable based on data shown in Figure 26.

Third; the rule editor is a core part of the fuzzy system in which all rules for building an efficient system are defined. As noted before, if-then rules are being applied to all defined variables and using logical operators, several variables can be involved in one rule. There is no limitation on defining rules but they should be defined in such a way to lead to optimum output. In addition, the parallel nature of the rules is one of the important aspects of the fuzzy system so the order of rules does not have any influence on the final output.

In our study, the mathematical equivalency of "*min*" and "*max*" were chosen for logical operators of "*And*" and "*Or*" respectively. Using six variables, seven equally weighted rules were defined as follows:

- *If (Point-Density is Birch) and (Hull-Ratio is Birch) and (Slope-Changes is Birch) then (Species is Birch) (1)*
- *If (Point-Density is Spruce) and (Hull-Ratio is Spruce) and (Slope-Changes is Spruce) then (Species is Spruce) (1)*
- *If (Point-Density is Pine) and (Hull-Ratio is Pine) and (Slope-Changes is Pine) then (Species is Pine) (1)*
- *If (Point-Density is Birch) and (Fitted-Cone is not Spruce) then (Species is Birch) (1)*
- *If (Point-Density is Spruce) and (Fitted-Cone is Spruce) and (Fitted-Cylinder is Spruce) and (Fitted-Sphere is Spruce) then (Species is Spruce) (1)*
- *If (Point-Density is Pine) and (Fitted-Cone is not Spruce) and (Fitted-Cylinder is Pine) and (Fitted-Sphere is Pine) then (Species is Pine) (1)*
- *If (Point-Density is not Birch) and (Slope-Changes is Pine) and (Fitted-Cylinder is Pine) and (Fitted-Sphere is not Birch) then (Species is Pine) (1)*

Based on the fuzzy inference system concepts in the Sugeno type for each rule, first, the inputs are being fuzzified, then fuzzy operations would be applied to the fuzzified inputs and the final output of the system would be the weighted average of all the rules outputs. This rule set, which comprises seven rules has been selected among twelve other sets, i.e. based on the final evaluation of the system, this set with this combination of rules has achieved a higher degree of accuracy amongst other sets.

3 Results and Discussions

3.1 Tree Detection

For the first part of this thesis work, three areas named *A*, *B* and *C* are involved. The first two areas (A and B) are large areas with the approximate size of 30 (m) * 70 (m), while area C is a strip with the width of 6 meters and length of 60 meters. According to the methodology part for doing analysis on large areas, they are divided into several strips with optimum width. Definition of the optimum width depends on the forest density and tree species. The average crown diameter of measured sample trees in the area, gives the strip width. Those strips for each area are named by indexes (A1, A2 …). It should be noted that for running the X-Y filtering algorithm on the so called areas, three different thresholds were used: for area A, the threshold of 2.0 meters was used because of more pine trees, area B is analysed with a threshold of 1.5 meters due to lots of spruce trees and finally 2.5 meters was used as the threshold for the X-Y filtering of area C because of birch and oak trees. Figure 28 shows the filtering result for area A.

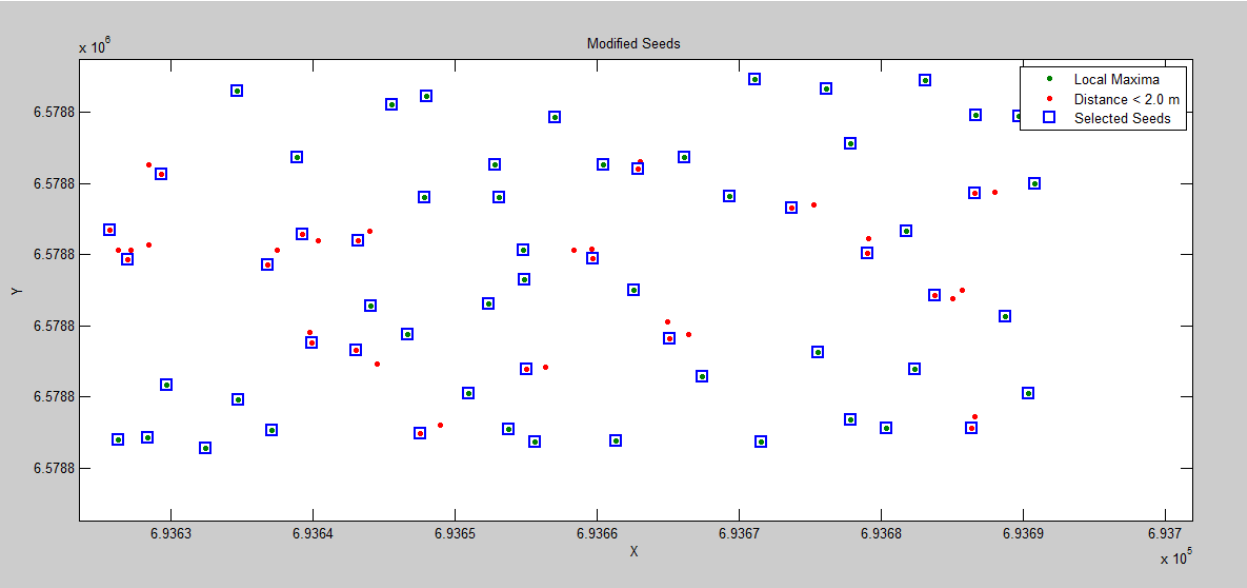
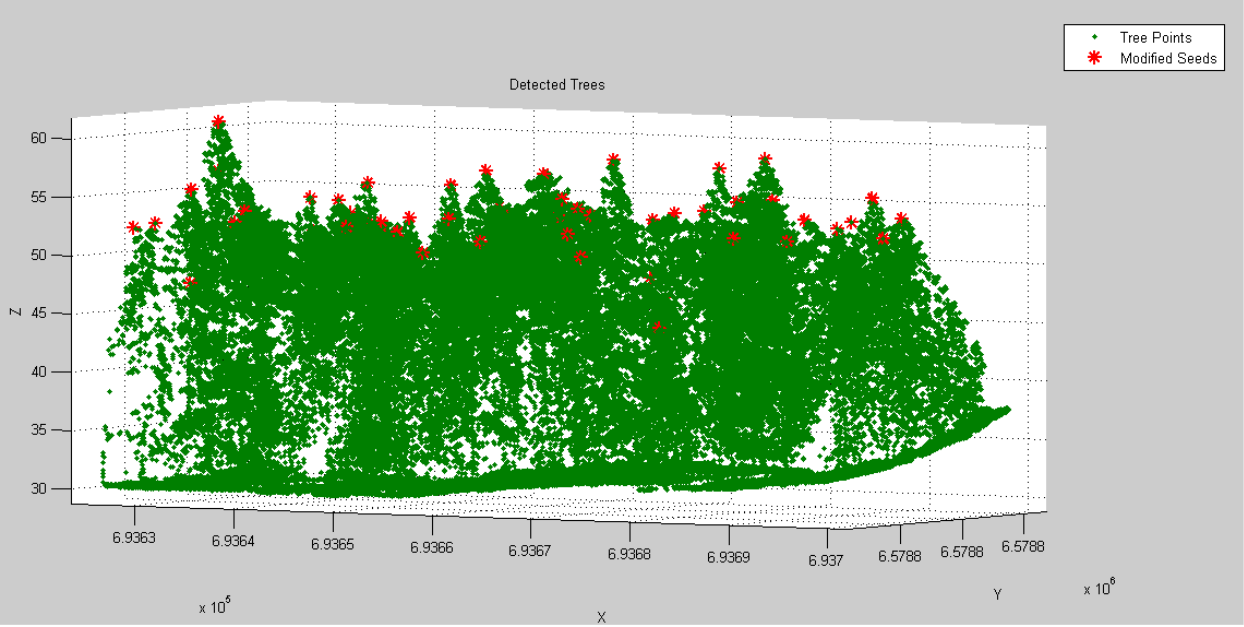
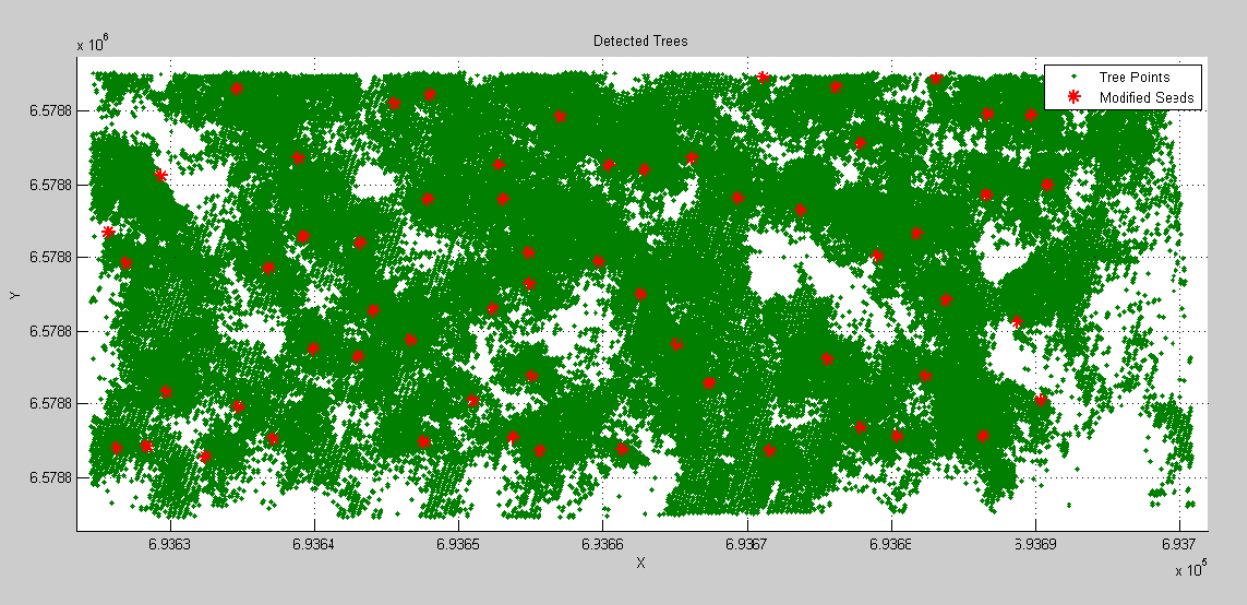


Figure 28: Accumulation of all local maxima points from all strips of area A in red and green points. Red points show the local maxima points, which are detected in a close distance of 2 meters to each other. Blue squares represent the result of filtering and chosen points as modified local maxima to use as external seed points in clustering.

In Figure 29, area A is shown with its detected trees in two different views for better visualization.



(a)



(b)

Figure 29: The detected trees in area A are shown as a red star, which represent the tree top points. Picture (*a*) is the side view of the area and picture (*b*) shows the top view of the area. It should be mentioned that red points are used as external seeds for clustering that is why they are called as modified seeds in the legend.

In this section the results of applied algorithms on data for tree detection, clustering procedures and the results of the validation part, are being discussed in three assessment types as follows:

3.1.1 User's accuracy

For evaluating the results and accuracy assessment, the modified seeds representing trees were imported into the Cyclone software where the trees were identified visually. The numbers of correctly detected trees were used for computing the user's accuracy percentage. Table 2 shows the result of this accuracy. The filtering processes were carried out on all indexed strips individually, so the total sum of detected trees in strips is not necessarily the same as detected trees in their corresponding area. In other words, every strip is treated individually for accuracy assessment. As can be seen in the table, the average accuracy is 93%. Due to lack of field measurements for determining reference trees, the calculation of producer's accuracy was not possible.

Generally, the ability of this method in finding the short trees, which are dominated by larger trees, can be considered as an advantage. As mentioned before in the other methods the tree canopy is analysed from top view, so many young or short trees are being hidden from investigation in dense forests. The other issue that should be noted is defining a threshold for the filtering analysis. It is clear from Table 2 and Table 3 that the algorithm could not find all trees in each strip. This problem can be explained by the threshold value. In dense forests with different species of trees, if the threshold value is chosen according to the crown diameter of older or wider trees, then the younger trees with narrow crowns would hide or delineate wrongly and vice versa. If the threshold is being determined based on narrow crown trees, then the other older trees would be detected as more than one single tree. Usually the threshold value for coniferous forests is 1.5-2.5 meters while for broadleaved trees or older trees is 2-5 meters, based on our own data and study. The problem of dense forest was also mentioned in works carried out by Woulder et al. (2000), Maltamo et al. (2004), Tiede et al. (2008) and Gupta (2010); they all have a consensus on limitation of using local maxima for crown delineation specially in dense and highly structured forest.

Table 2: Number of detected trees in different areas and strips with computed accuracy based on correctly detected trees.

Area Name	*No. of Detected Tree Top Points*	*No. of Correctly Detected*	*User's Accuracy*
A	61	54	88%
A1	11	9	82%
A2	13	12	92%
A3	13	13	100%
A4	16	14	87%
A5	16	14	87%
B	95	90	95%
B1	20	19	95%
B2	22	22	100%
B3	26	24	92%
B4	22	21	95%
B5	13	13	100%
C	11	11	100%
Total average of User's Accuracy			93%

3.1.2 DPM comparison

In the second type of the assessment, our results are compared with the DPM results. The comparison was divided into two classes: *exact* and *nearly exact*. Class *exact*, means that the detected trees coincide exactly with the trees detected by DPM. Class of *nearly exact* covers the detected trees in 2.5 meters buffer zones of detected trees by DPM. The rest of the trees belonging to the range of more than 2.5 meters distance are not involved in our accuracy assessment. The computed accuracy shows the percentage of matching trees between the DPM and the thesis method in both classes of *exact* and *nearly exact*. The overall accuracy is computed as 85% for our trees, which match with the DPM result. In the other step, the number of trees in each area is counted visually in Cyclone so as it can be seen in Table 3, the overview comes from the number of detected trees from the thesis method, DPM and Cyclone. Based on the statistics of Table 2 and Table 3, it can be computed that the developed algorithm can reach up to 90% accuracy in detecting trees. This percentage shows the total accuracy, which comes from dividing correctly detected trees from Table 2 by visually inspected and detected trees from Cyclone (Table 3) for each strip. Statistics in detail are expressed in Table 3. Figure 30 shows the areas with detected tree points in Quantum GIS software.

Table 3: The number of detected trees from thesis method, DPM software and visually detected from Cyclone. This table also shows an accuracy of matching detected trees from DPM and thesis method.

Area Name (Strip)	*No. of Detected Trees*			*Comparison with DPM*		
	Thesis	*DPM*	*Visually by Cyclone*	*Exact*	*Nearly Exact*	*Accuracy*
A	61	51	62	24	14	74%
A1	11	12	11	1	5	50%
A2	13	10	14	6	1	70%
A3	13	14	15	8	1	64%
A4	16	9	17	6	3	100%
A5	16	6	14	4	2	100%
B	95	56	108	39	12	91%
B1	20	7	16	5	2	100%
B2	22	11	24	9	1	91%
B3	26	17	32	12	4	94%
B4	22	14	23	14	0	100%
B5	13	7	14	2	4	86%

It is obvious from the above table that the number of detected trees from the thesis method is quite close to the number of visually detected trees, and in some areas such as B, there is a significant difference between the detected number of trees in DPM and Cyclone while the thesis method has given a better result. It should be noted that both areas of A and B have around 2000 m^2 area while B has 108 trees versus 62 trees of A, which confirms the high density of B.

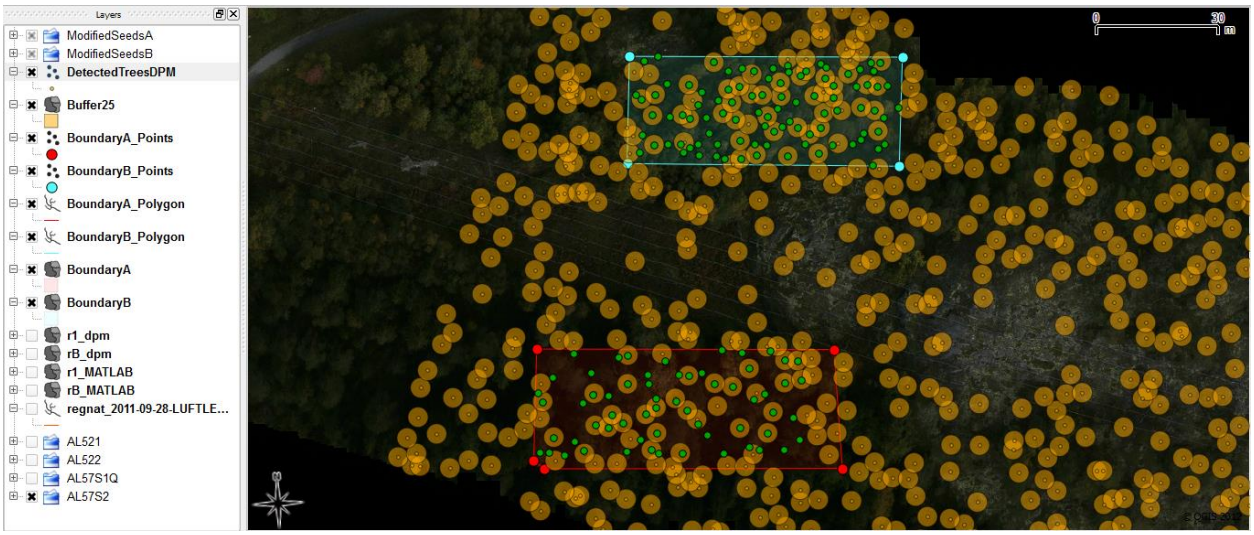


Figure 30: Area B is shown by a blue boundary at top of the picture and area A is shown with red boundary below it. The green points are detected trees using the thesis method while the small orange points are showing the trees from DPM and orange circles around those orange points define 2.5 meters buffer zone of those trees.

3.1.3 Comparison of clustering methods

Three types of clustering methods were applied on all of the strips to get a better overview of these method's results on real data. The two areas of A and B are not involved in this experiment because for running the hierarchical method on a large data set, it is required very large amount of memory, which is not available in my device. Results of all three methods are compared with each other based on the silhouette value, which is a measure of clustering goodness. These results are written in Table 4.

Table 4: Mean silhouette value for all clusters of each strip is obtained for all three methods. Analysis is performed on top 50 percent of tree points.

Area Name (Strip)	*Mean Silhouette Value*		
	Unsupervised K-means	*Supervised K-means*	*Hierarchical*
A1	0.6212	0.6198	0.5187
A2	0.5034	0.4715	0.3335
A3	0.5258	0.5191	0.4814
A4	0.5368	0.5157	0.3764
A5	0.5128	0.5152	0.3839
B1	0.5014	0.4717	0.3795
B2	0.5019	0.4881	0.3291
B3	0.4406	0.4532	0.3097
B4	0.4478	0.4457	0.2986
B5	0.5224	0.5615	0.3844
C	0.6343	0.6573	0.6002

Unsupervised K-means

In the first experiment, the normal K-means method was applied on the upper section (50 percent) of tree points without any external seeds, so the K-means based on its mathematical structure has performed clustering to reach desired number of clusters, which is a common input to all three methods. It is clear from Table 4 that the average amount of silhouette value for the unsupervised method is more than the value of other methods but it does not necessarily mean better clustering of trees due to the mathematical nature of the silhouette value. Normal K-means treats the points as a bunch of objects for clustering and as it does not have any pre-defined initial points for starting the clustering, so in each iteration, it starts with a new set of initial points. For instance in five iterations, it computes five different "sum of distances" values and therefore it gives five different cluster centers. It is crystal clear that the value of silhouette would be changed in each iteration (the silhouette value in Table 4 comes from the first iteration). Furthermore, the result of unsupervised clustering for one tree-area is shown in Figure 31(a) and easily the difference of results can be visualized. As a conclusion, this method is not reliable for clustering trees in comparison with supervised K-means.

Supervised K-means

Opposite to the previous method, in the supervised K-means the position of initial seeds, cluster centroids and sum of distances are fixed and do not change during replications. In this method the number of clusters and position of initial points are given as seed points, it was hypothesized that due to the use of external seed points, K-means gives better clustering result; the output of this method is shown in Figure 31(b), which can be compared with other results. In the case of the other strips after careful visual investigation, it is found that this method works better in clustering. As known from previous studies and the concept of K-means clustering, it is obvious that all clustering methods especially K-means, work well for compact and isolated ALS data. It is noticed from the results that in dense forests, where the trees are close to each other, the cluster of one tree spreads to the neighbouring trees, so for decreasing this effect other experiments were carried out by reducing the height of the data zone from 50% to 40%. In this case, although the better silhouette values were obtained as well as better grouping, some of the trees with low height were ignored from the clustering process.

Hierarchical Clustering

Referring to Table 4, the silhouette value for hierarchical clustering is smaller than the other two methods. It is also clear from the visual inspection in Figure 31, that the hierarchical method works more efficiently than unsupervised K-means in tree clustering. In less dense areas with isolated trees, the hierarchical approach yields the same result as the supervised K-means. The main drawback of the hierarchical method is the computational time. Because of creating a dendrogram, which links all points to each other, the hierarchical method requires huge memory for clustering. As we usually deal with large amount of airborne data in forestial analysis, this method would not be as efficient as K-means. The result of clustering with this method is illustrated in Figure 31(c).

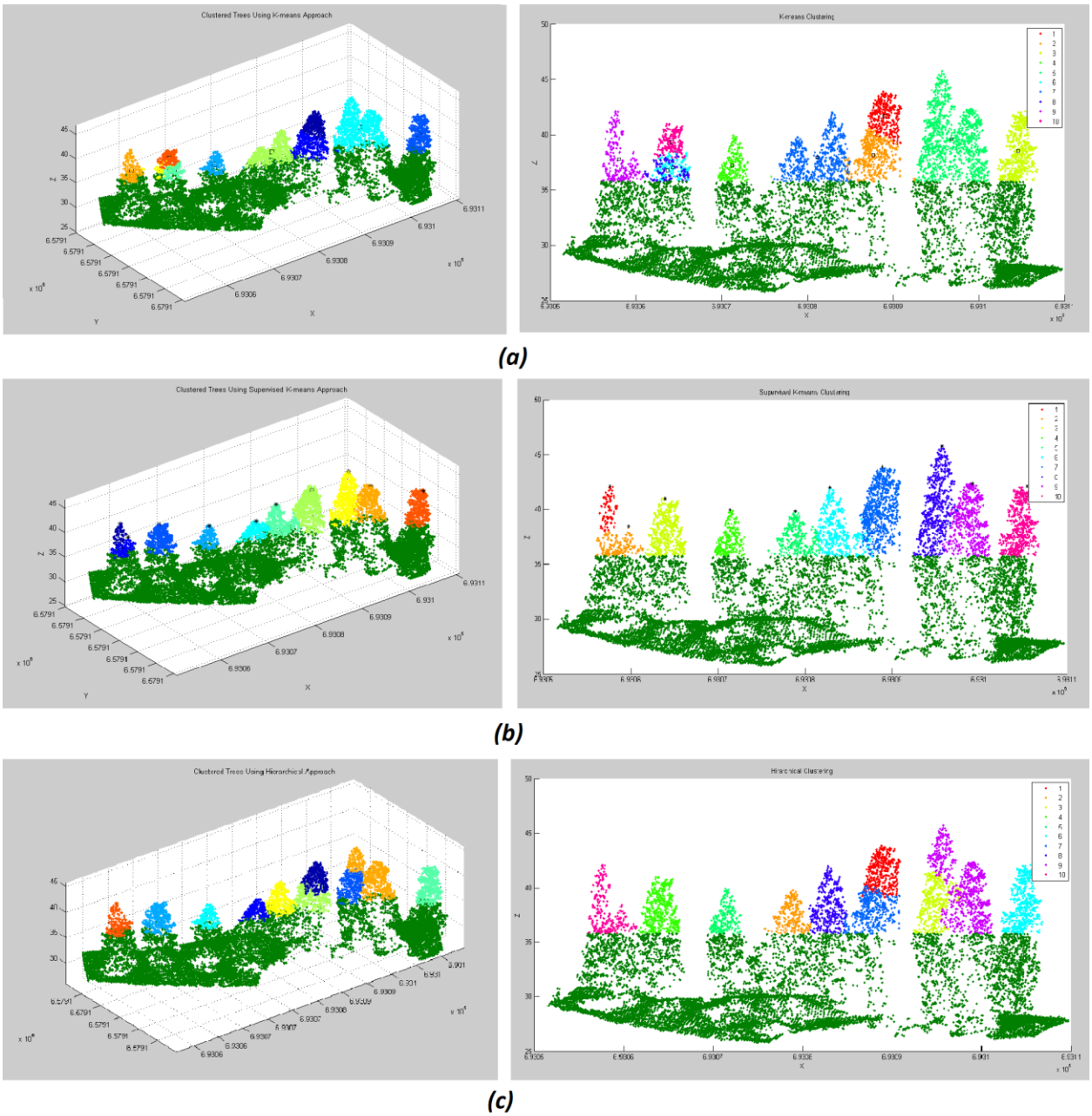


Figure 31: The first set of pictures *(a)* shows the result of clustering for the *Unsupervised K-means* method. In each set, the right picture shows a 2D scatterplot versus left picture, which shows a 3D plot of same area, dark green points belong to ground and part of trees, which are not involved in clustering. In set (*a*), the hollow squares illustrate the centre points of clusters. Set *(b)* shows the result of *supervised K-means*, the star points on top of each tree represent external seed points for clustering initializing. Two pictures of set *(c)* show the clustering result of *Hierarchical approach*.

It is obvious that based on the clustering results from the above figure and mathematical concept of each method, the supervised K-means approach is more efficient than other methods. The result of supervised K-means, shows only few points that were grouped from neighbouring trees as one individual tree due to very close distance of those trees.

3.2 Tree Species Identification

For performing the second part of this study, the species of 118 trees were identified and corresponding LiDAR data were extracted from the point clouds. All developed methods were tested on individual trees and a specific value from each method was saved for further computations. Three standard error values of fitting different shapes, point density ratio, hull ratio and number of slope changes were saved for each tree as specification of that tree for the evaluation step. As mentioned in the previous section, the range and histogram-shape of results of each method provide necessary information for defining the membership functions and finally running the fuzzy system. Looking on Figure 32, all defined MFs for variables can be seen. Based on those MFs, seven rules were developed and the result of the rules aggregation is represented in the Rule Viewer window in Figure 33. The Rule Viewer window displays a road map of the whole fuzzy inference process. In other words, all the concepts of FIS are shown in this window, from fuzzifying the inputs to defuzzifying the output. This window provides us with a final defuzzified value of output for any input values, with sliding red lines for each input or inserting different input values we can see the changes of the defuzzified number. As we defined our output as constant parameters of 1, 2 and 3 for birch, spruce and pine respectively, it is obvious to obtain a final result in range between 0 and 3. With a simple condition, which is coded, the species of tree are analysed and computed as constant numbers by entering the six specification values of one tree.

1. $if\ round(final\ output) = 1 \overset{then}{\Longrightarrow} final\ species = Birch$
2. $if\ round(final\ output) = 2 \overset{then}{\Longrightarrow} final\ species = Spruce$
3. $if\ round(final\ output) = 3 \overset{then}{\Longrightarrow} final\ species = Pine$

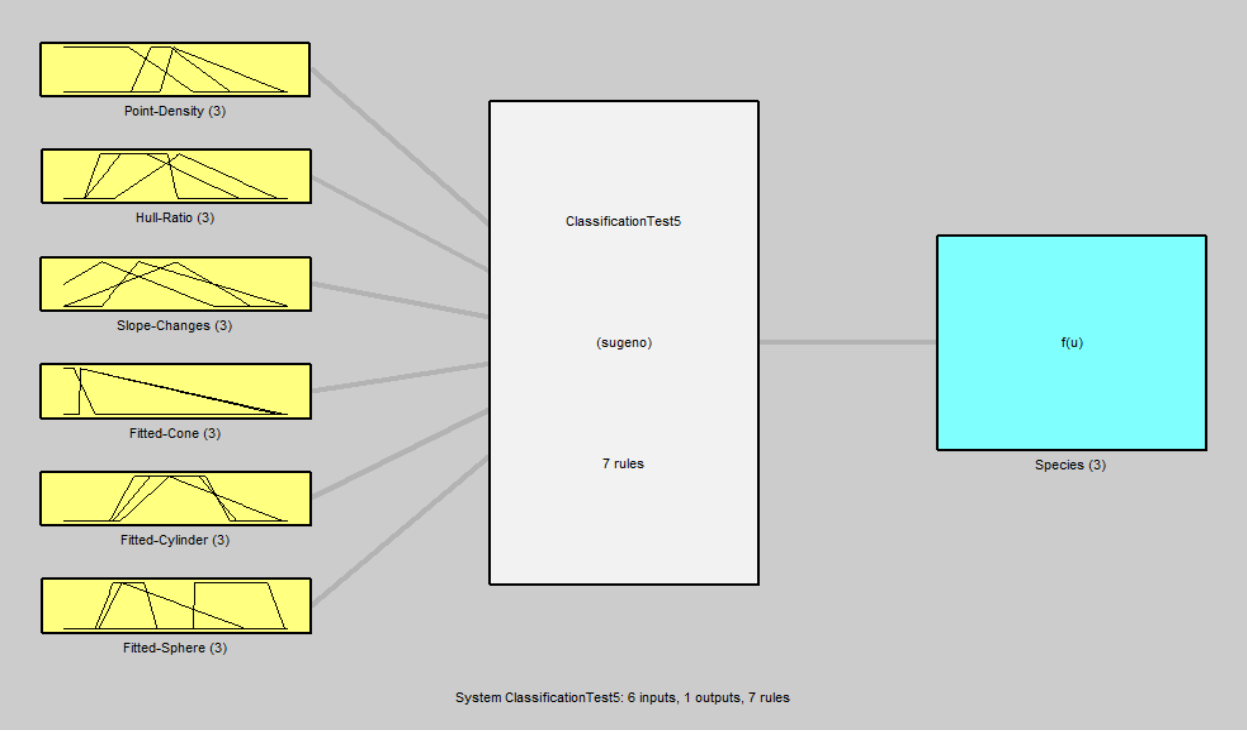


Figure 32: All defined MFs for each variable are shown in this image as well as the FIS type and number of rules.

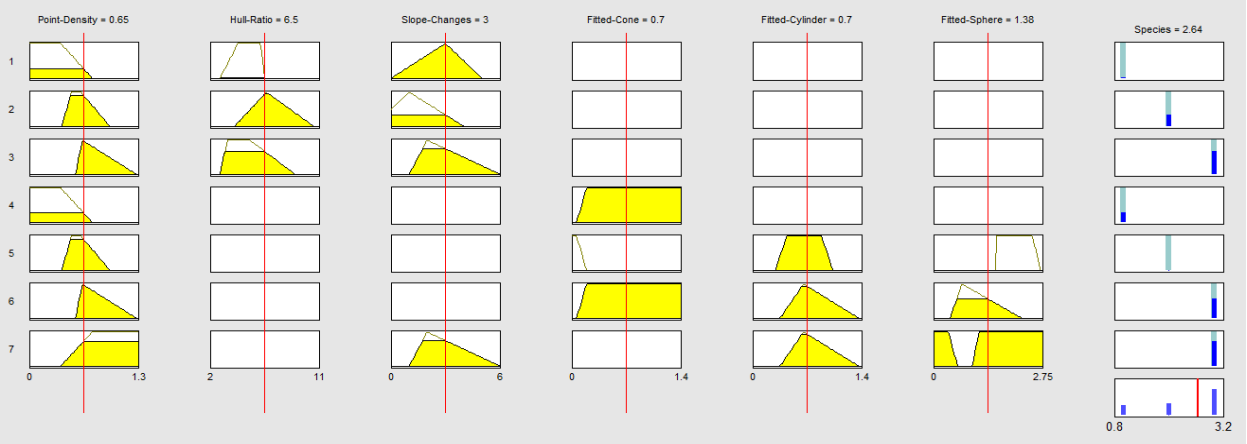


Figure 33: The Rule Viewer window in which six left columns representing input variables and seven rows illustrating the seven defined rules. Each rectangle in the right column shows the implicated result of that row (rule) and finally the lower rectangle depicts the aggregated result, graphically and the numerical result is shown on top of that column. By sliding the red line, one can see the changes in both numerical and graphical final result. Blank rectangles in each row show variables, which are not involved in that rule.

3.2.1 Model validation

The Fuzzy inference system is expected to provide us with an optimum model, which aggregates the result of all methods and yields an as accurate as possible output. For validating the model, the reference trees are evaluated by the current model. The evaluation experiment is carried out in two levels. In the first level, the trees are divided into three species of birch, spruce and pine for evaluating the number of correctly identified species. In the second level, the trees are classified more generally to coniferous and deciduous species. As most of the previous studies were carried out on classifying coniferous-deciduous types, the second level of evaluation was performed to obtain a value for comparing. The results of these experiments are shown in Table 5 and Table 6.

Table 5: The confusion matrix shows the percentage and number of classified species; totally 33 birch, 42 pine and 38 spruce trees were involved in the accuracy assessment experiment.

Classified	*Birch*		*Norway spruce*		*Scots pine*		*Overall Accuracy*
	No.	*%*	*No.*	*%*	*No.*	*%*	
Birch	24	73	3	8	1	2	
Norway spruce	4	12	33	87	11	26	
Scots pine	5	15	2	5	30	72	
Proportion correct	73 %		87 %		71 %		77 %

Table 6: The percentage and number of classified coniferous-deciduous species are shown.

Classified	*Deciduous*		*Coniferous*		*Overall Accuracy*
	No.	*%*	*No.*	*%*	
Deciduous	24	73	4	5	
Coniferous	9	27	76	95	
Proportion correct	73 %		95 %		84 %

It is obvious from Figure 32 that none of the methods can succeed in partitioning completely. However, from Table 5, it can be interpreted that the aggregation of the methods by the fuzzy inference system resulted to overall 77% of accuracy. Although evaluating this model by hundreds of trees may affect the overall result, it should be considered that more samples would define better and more accurate membership functions.

Table 5 also reveals the fact that spruces are classified more accurately; this can be proved by referring to MFs and results of the methods, because the cone and sphere fitting approaches were very efficient in detecting spruce types as can be seen in Figure 32. In the same picture it is shown that cone fitting, could not recognize pine and birch trees so their MFs overlapped each other while the point density method can define a rather distinct range for birch and pine trees.

The worth case was taken place in the cylinder fitting method that despite the expectation of obtaining better result for pine trees, which supposed to have cylindrical shape nature, all three types were laid in the same ranges of the standard error aspect.

The hull ratio method was rather efficient in spruce detection and the outcome of the slope changes approach was used as an auxiliary factor for determining spruce and pine trees.

These results from only LiDAR points express the fact that by using other data sources of remote sensing such as spectral images, aerial photos and thermal images, it is decisive to attain very high accuracy in the field of tree species classification.

3.2.2 Applying obtained model on test area

As a final step in this study, the obtained model from the fuzzy system, which provides us with the mentioned accuracy results in Table 5 and Table 6 is applied to one of the test areas. In this test area, the number of trees has been detected already by using the algorithms explained in section 2.2. Using the detected tree top points as initial seeds for clustering, the group of laser points, which belong to each tree is extracted from the test site. It should be noted that the optimum clustering results were obtained using *supervised K means* algorithm as explained in section 2.3.2. The extracted tree clusters are used as input data for running the algorithms of six methods for tree species identification. As a final result those six variables for each tree are implemented in a fuzzy model and the corresponding species of any tree were determined. In Figure 34, each species are represented by specific colour for better visualization. The accuracy of detected trees is being investigated in section 3.1 for this test site, but due

to lack of information about tree species in so called area, the accuracy assessment of the determined species are not possible.

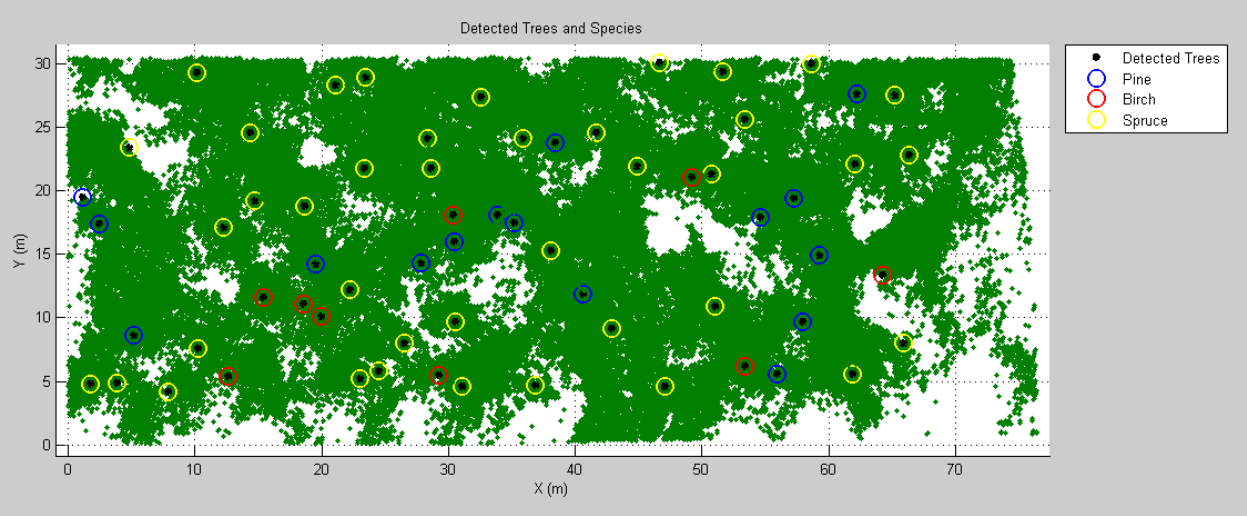


Figure 34: Detected trees and their species are shown for test area A. The green points are raw laser points, which illustrate tree and ground points together. Black points represent detected trees in this area and blue, red and yellow circles around those black points, show the species of each detected tree as pine, birch and spruce respectively.

The accuracy of determining tree species in dense forestial areas can be influenced by three factors. Based on developed algorithms in this study, first of all it is inevitable to face with error in detecting correct tree top points, clustering may cause the second level of error in this approach in a case that some irrelevant points being clustered as one single tree points, which consequently would affect the result of six algorithms of species identification. As a third source of error, the fuzzy model can be outlined, as mentioned the obtained model does not provide 100% accuracy for species distinguishing. Another point that should be emphasized is that due to dividing species to three classes of birch, spruce and pine, it is probable to find some other species, which are identified as one of those mentioned species, however the portion of other species trees may not exceed a few percent in a large forest area due to the nature of the Swedish forests.

4 Conclusions and future works

Using LiDAR data for forestial areas and developing some algorithms, which use such kind of data, could end up in good results in determining the forest inventories. This thesis work was performed based on LiDAR data to find a rapid and accurate solution for determining two features of forest inventories: the number of trees and their species. As can be noticed from previous studies, a considerable progress has been achieved in this field using different methods. In this study, most concentration was given to laser point data. Using this type of data, a bunch of algorithms were written and developed by the author to detect trees and identify their species in forests.

In general, the main factors that have most influence in determining the number and shape of trees are the LiDAR point density, the forest conditions in which tree grows, the terrain type, the crown cover and the tree density. Usually, outliers create a problem in tree detection, removal of which from the raw data significantly can improve the algorithm performance and output quality (Gupta, 2010). Apart from these parameters, the impact of data acquisition time should not be forgotten; in other words, separate processing of leaf-on and leaf-off data can lead to better results in tree estimation and species discrimination.

As a conclusion for the first part of this study, which is conducted by single tree detection algorithms, the impact of the threshold value in the filtering process, should be noted. Setting the threshold distance becomes more difficult with increasing forest complexity. Normally, this value is lower for younger trees with narrow crown at the top compared to relatively older trees with wider crown and intermittent peaks at the top. Therefore, the significance of pre-knowledge about test areas should not be ignored due to its influence on setting variables. In case of clustering methods, which two main approaches were tested in this study, the traditional *K*-means method generates bad clustering because of its randomized process. The hierarchical approach has a good result in grouping when it has been applied to set of data in which trees are separate from each other. The supervised *K*-means algorithm is comparatively good for portioning the objects when the user has control over the seeding, so the tree top points, which were detected as local maxima are used as an external seed points for performing supervised *K*-means algorithm in this study. Finally comparing obtained results for detected trees with the results of DPM software for the same area and achieving more than 85% precision was an encouraging result for the developed algorithms in this study.

Species identification of detected trees was carried out in this study based on a model obtained from a fuzzy logic system, which aggregated the result of six different methods for defining an efficient model. Those six applied methods, investigated tree shapes from a different point of view, so aggregating the results is a matter of consideration. In the rule defining step of the fuzzy system, it is important that the rules are being well-defined because of its direct effect on the validation step of the model, so careful inspecting of the membership functions of each method can give the user more knowledge for

performing this step (defining rules). For identifying species, more reference trees would lead to a more precise model due to the statistical reasons that a larger sample of data provides more confident results.

Finally, it should be mentioned that the obtained results present the accuracies of detected trees and their identified species in the limited number of test areas with few determined reference trees and using only laser point data as input of algorithms. It is crystal clear that these results could not reflect the best results in this field (tree detection and species identification), but can be considered as a motivation for expanding developed algorithms by using other sources of data such as spectral images, intensity values, thermal images, etc., and applying them on different forestial areas for identifying more species rather than three specific types. As mentioned before, these algorithms were developed for applying on Swedish forests, which are mainly covered by coniferous types, therefore using variate data sources, especially images would make it possible to run our algorithms on tropical forests also.

In a further step, automatic classifying of all natural and man-made objects from LiDAR data and images would be considered as a future attempt for developing this study.

5 Bibliography

Adams, R.A., 1990. *Calculus*. 33rd ed. Addison-Wesley Publishers.

Brandtberg, T. & Warner, T., 2006. High-spatial-resolution remote sensing. In Shao, G. & Reynolds, K.M., eds. *Computer applications in sustainable forest management*., 2006. Springer.

Dhillon, I. & Modha, D., 2001. Concept decompositions for large sparse text data using clustering. In D. Fisher, ed. *Machine Learning*. Kluwer Academic Publishers. pp.143-75.

Fan, H., 2010. *Theory of Errors and Least Squares Adjustment*. Stockholm: Universitetsservice AB.

Forina, M., Armanino, C. & Raggio, V., 2002. Clustering with dendrograms on interpretation variables. In *Analytica Chimica Acta*. Genova: Elsevier. pp.13-19.

Gupta, S., 2010. *Single tree detection and modeling using airborne laser scanner data*. PhD Thesis. Freiburg im Breisgau, Germany: Albert-Ludwigs-Universität.

Han, J. & Kamber, M., 2001. *Data Mining: Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann Publishers.

Heinzel, J.N., Weinacker, H. & Koch, B., 2008. Full automatic detection of tree species based on delineated single tree crowns – a data fusion approach for airborne laser scanning data and aerial photographs. In Hill, R.A., Rosette, J. & Suárez, J., eds. *Proceedings of SilviLaser 2008: 8th international conference on LiDAR applications in forest assessment and inventory*. Edinburgh, 2008.

Hill, R.A. & Thomson, A.G., 2005. Mapping woodland species composition and structure using airborne spectral and LiDAR data. *International Journal of Remote Sensing*, 26, pp.3763-79.

Hollaus, M. et al., 2009. Growing stock estimation for alpine forests in Austria: A robust LiDAR-based approach. *Can. J. Forest Res.*, 39, pp.1387-400.

Holmgren, J., Persson, A. & Soderman, U., 2008. Species identification of individual trees by combining high resolution LiDAR with multi-spectral images. *International Journal of Remote Sensing*, 29, pp.1537-52.

Horn, H.S., 1971. *The Adaptive Geometry of Trees*. Princeton: Princeton University Press.

Hyyppä, J. et al., 2008. Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests. *International Journal of Remote Sensing*, 29, pp.1339-66.

Jain, A.K., Murty, M.N. & Flynn, P.J., 1999. Data Clustering: A Review. *ACM Computing Surveys*, 31(3), pp.264-323.

Koch, B., Heyder, U. & Weinacker, H., 2006. Detection of individual tree crowns in airborne lidar data. *Photogrammetric Engineering and Remote Sensing*, 72, pp.357-63.

Korpela, I., 2004. *Individual tree measurements by means of digital aerial photogrammetry*. PhD Thesis. Helsinki: Silva Fennica Monographs University of Helsinki, Department of Forest Resource Management.

Ko, C., Sohn, G. & Remmel, T.K., 2009. Classification for deciduous and coniferous trees using airborne LiDAR and internal structure reconstructions. In *Proceedings of SilviLaser*. College Station, Texas, 2009.

Koukoulas, S. & Blackburn, G.A., 2005. Mapping individual tree location, height and species in broadleaved deciduous forest using airborne LIDAR and multi-spectral remotely sensed data. *International Journal of Remote Sensing*, 26, p.431–455.

Lämås, T. & Eriksson, L.O., 2003. Analysis and planning systems for multiresource, sustainable forestry: The Heureka research programme at SLU. *Canadian Journal of Forest Research*, 33, pp.500-08.

Liang, X., Hyyppä, J. & Matikainen, L., 2007. Deciduous-coniferous tree classification using difference between first and last pulse laser signatures. *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, 36, pp.253-57.

Magnusson, M., 2006. *Evaluation of remote sensing techniques for estimation of forest variables at stand level*. PhD Thesis. Umeå: Acta Universitatis Agriculturae Sueciae Swedish University of Agricultural Sciences.

Maltamo, M. et al., 2006. Nonparametric estimation of stem volume using airborne laser scanning, aerial photography and stand-register data. *Canadian Journal of Forest Research*, 36, pp.426-36.

Maltamo, M. et al., 2004. The accuracy of estimating individual tree variables with airborne laser scanning in boreal nature reserve. *Canadian Journal of Forest Research*, 34, pp.1791-801.

Maltamo, M. et al., 2007. Experiences and possibilities of ALS based forest inventory in Finland. In Rönnholm, P., Hyyppä, H. & Hyyppä, J., eds. *Proceedings of ISPRS Workshop on Laser Scanning 2007and SilviLaser 2007*. Espoo, 2007.

Maltamo, M. et al., 2009. Combining ALS and NFI training data for forest management planning: a case study in Kuortane, Western Finland. *European Journal of Forest Research*, 128, pp.305-17.

Mao, J. & Jain, A.K., 1996. A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Trans Neural Netw.*, 7, pp.16-29.

McQueen, J., 1967. Some methods of classification and analysis of multivariate observations. In Le Cam, L.M. & Neyman, J., eds. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, CA, 1967.

Næsset, E., 2002. Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data. *Remote Sensing of Environment*, 80, pp.88-99.

Nelson, R.F., Latty, R.S. & Mott, G., 1984. Classify Northern Forests using Thematic Mapper Simulator Data. *Photogrammetric Engineering and Remote Sensing*, 50, pp.607-17.

Nilsson, M., 1996. Estimation of tree heights and stand volume using an airborne LiDAR system. *Remote Sensing of Environment*, 56(1), pp.1-7.

Ørka, H.O., Næsset, E. & Bollandsås, O.M., 2009. Classifying species of individual trees by intensity and structure features derived from airborne laser scanner data. *Remote Sensing of Environment*, 113(6), pp.1163-74.

Ørka, H.O., Næsset, E. & Bollansås, O.M., 2007. Utilizing airborne laser intensity for tree species classification. *ISPRS Workshop on Laser Scanning and SilviLaser*, 36, pp.300-04.

Packalén, P., 2009. *Using airborne laser scanning data and digital aerial photographs to estimate growing stock by tree species*. PhD Thesis. Joensuu: University of Joensuu, Faculty of Forest Sciences.

Persson, Å., Holmgren, J. & Söderman, U., 2002. Detecting and measuring individual trees using an airborne laser scanner. *Photogrammetric Engineering and Remote Sensing*, 68, pp.925-32.

Persson, Å., Holmgren, J. & Söderman, U., 2006. Identification of tree species of individual trees by combining very high resolution laser data with multi-spectral images. *Workshop on 3D Remote Sensing in Forestry*, pp.91-96.

Popescu, S.C., Wynne, R.H. & Scrivani, J.A., 2004. Fusion of small-footprint LiDAR and multispectral data to estimate plot-level volume and biomass in deciduous and pine forests in Virginia. *USA. Forest Science*, 50, pp.551-65.

Reitberger, J., Krzystek, P. & Stilla, U., 2006. Analysis of full waveform data for tree species classification. *Symposium of ISPRS Commission III Photogrammetric Computer Vision PCV*, pp.228-33.

Reitberger, J., Krzystek, P. & Stilla, U., 2009. Benefit of airborne full waveform LiDAR for 3D segmentation and classification of single trees. In *ASPRS Annual Conference*. Baltomore, Maryland, 2009.

Rokach, L. & Maimoon, O., 2005. Clustering Method. In O. Maimon & L. Rokach, eds. *The Data Mining and Knowledge Discovery Handbook*. 1st ed. pp.321-52.

Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, pp.53-65.

Sneath, P. & Sokal, R., 1973. *Numerical Taxonomy*. San Francisco, CA: W.H. Freeman Co.

Suárez, J., Rosette, J., Nicoll, B. & Gardiner, B., 2008. A practical application of airborne LiDAR for forestry management in Scotland. In *Proceedings of SilviLaser*. Edinburgh, 2008.

Tiede, D., Hochleitner, G. & Blaschke, T., 2005. A full GIS-based workflow for tree identification and tree crown delineation using laser scanning. In Stilla, U., Rottensteiner, F. & Hinz, S., eds. *CMRT05. IAPRS*. Vienna, 2005.

Vauhkonen, J., 2010. *Estimating single-tree attributes by airborne laser scanning: methods based on computational geometry of the 3-D point data*. PhD Thesis. Joensuu: University of Eastern Finland, School of Forest Sciences, Faculty of Science and Forestry.

Wang, Y., 2009. *Fully automatic reconstruction of virtual environment based on LiDAR data*. PhD Thesis. Freiburg: Faculty of Forest and Environmental Sciences Albert-Ludwigs-Universität, Department of Remote Sensing and Land Information Systems.

Wikipedia: the free encyclopedia, n.d. [Online] Available at: http://www.en.wikipedia.org/wiki/Least_squares#Differences_between_linear_and_non-linear_least_squares [Accessed 16 November 2012].

Woodget, A.S., Donoghue, D.N.M. & Carbonneau, P., 2007. An assessment of Airborne LiDAR for Forest Growth Studies. *Ekscentar*, 10, pp.47-52.

Wulder, M., Niemann, K.O. & Goodenough, D.G., 2000. Local maximum filtering for the extraction of tree locations and basal area from high spatial resolution imagery. *Remote Sensing of Environment*, 73, pp.103-14.

Reports in Geodesy and Geographic Information Technology

The TRITA-GIT Series - ISSN 1653-5227

2012

12-001 **Atta Rabbi& Epameinondas Batsos.** Clustering and cartographic simplification of point data set. Master of Science thesis in Geoinformatics. Supervisor: Bo Mao. February 2012.

12-002 **Johanna Löfkvist.** Höjdmodellering med laserdata: Studie av Kärsön, Ekerö med fokus på upplösning, datalagring samt programvara. Master of Science thesis in Geoinformatics. Supervisor: Yifang Ban, KTH & Gabriel Hirsch, Sweco Position. February 2012.

12-003 **Mandana Mokhtary.** Sensor Observation Service for Environmental Monitoring Data. Master of Science thesis in Geoinformatics. Supervisor: Yifang Ban. May 2012.

12-004 **Dong Fang.** Moving Object Trajectory Based Spatio-Temporal Mobility Prediction. Master of Science thesis in Geoinformatics. Supervisor: Gyözö Gidofalvi. July 2012.

12-005 **Yezeed Abdelmajid.** Investigation and comparison of 3D Laser scanning software packages. Master of Science thesis in Geodesy No. 3126. Supervisor: Milan Horemuz. August 2012.

12-006 **Ana Fernández Torralbo & Pablo Mazuelas Benito**. Landsat and MODIS Images for Burned Areas Mapping in Galicia, Spain. Master of Science thesis in Geoinformatics. Supervisor: Yifang Ban. September 2012.

12-007 **Arfan Sohail.** Landuse and Coastline Change in the Eastern Mekong Delta (Viet Nam) from 1989 to 2002 using Remote Sensing. Master of Science thesis in Geoinformatics. Supervisor: Hans Hauska. October 2012.

12-008 **Maria Bobrinskaya.** Remote Sensing for Analysis of Relationships between Land Cover and Land Surface Temperature in Ten Megacities. Master of Science thesis in Geoinformatics. Supervisor: Yifang Ban. December 2012.

2013

13-001 **Mohammad Amin Alizadeh Khameneh.** Tree Detection and Species Identification using LiDAR Data. Master of Science thesis in Geodesy No. 3127. Supervisor: Milan Horemuz. January 2013.

TRITA-GIT EX 13-001
ISSN 1653-5227
ISRN KTH/GIT/EX--13/001-SE